EECE 5639 Computer Vision I

Lecture 8

- **Corners, Sift Features**
- **Project 2 is out.**
- **First Midterm: February 12 (Hw 1 & 2 ; Project 1)**

Next Class

- **Midterm Review**

Point Features
Corners

Corner Features

Places where TWO strong edges meet.

They can be used for:

- Object tracking
- 3D triangulation (stereo)
- Object recognition

Corners are useful

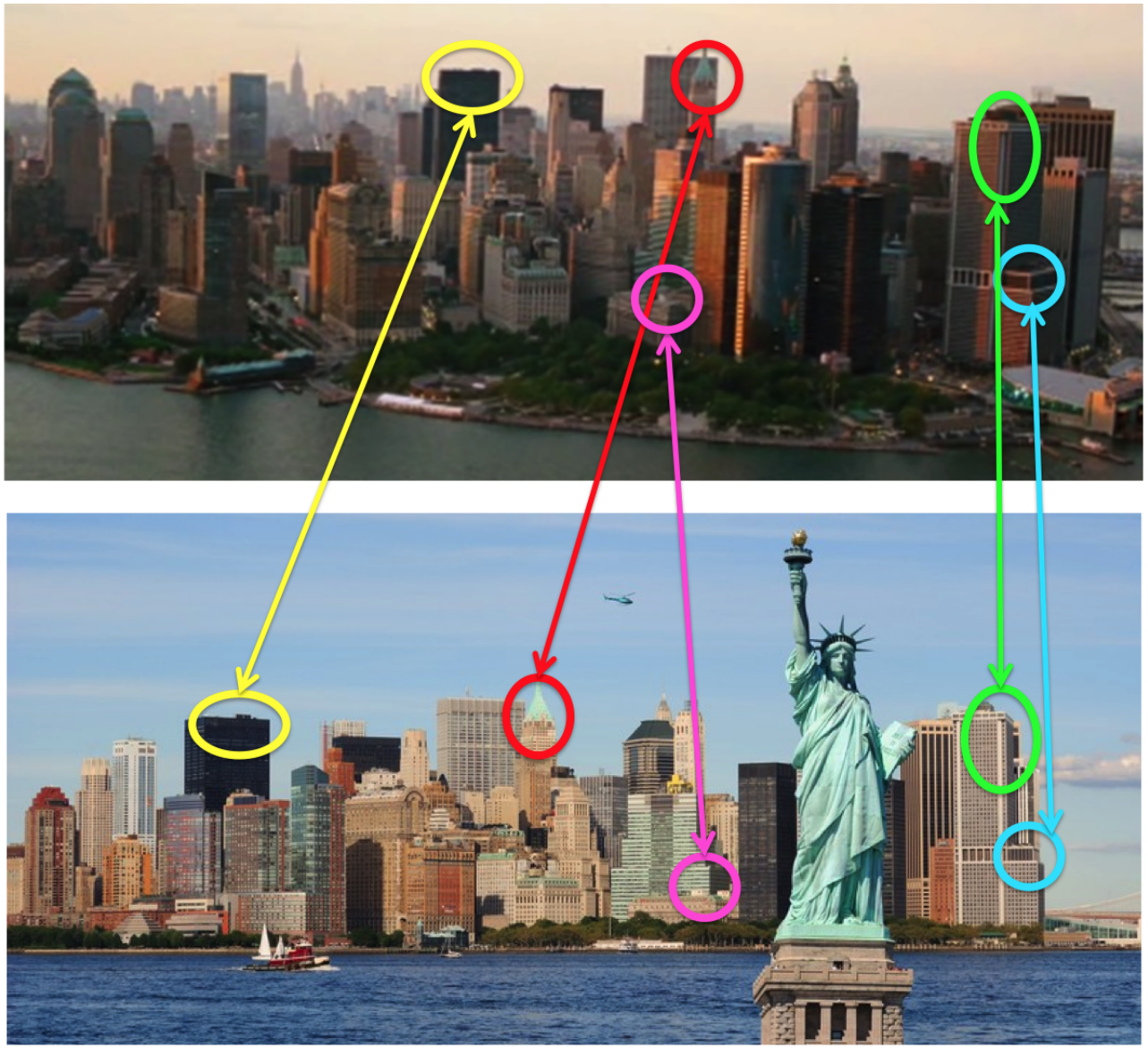
Which city?

Corners are useful

Which city?

Corners are useful

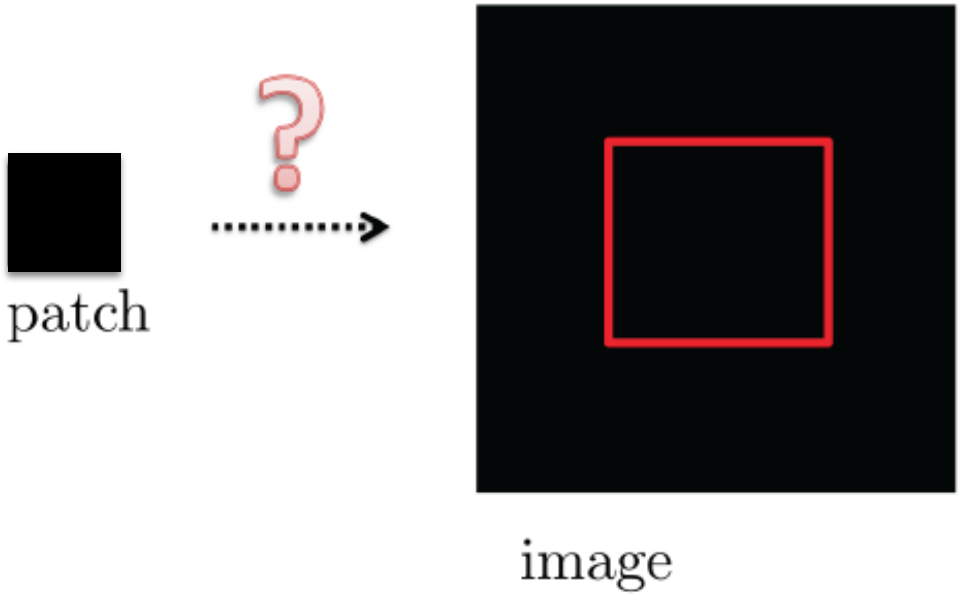
Which city?



Corners are useful

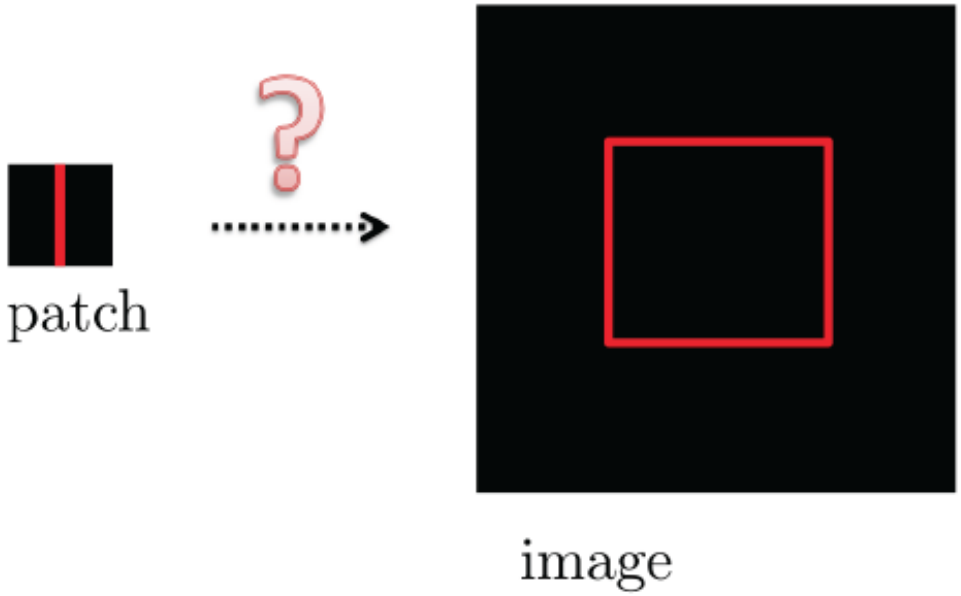
Where can I find this pattern?

Corners are useful



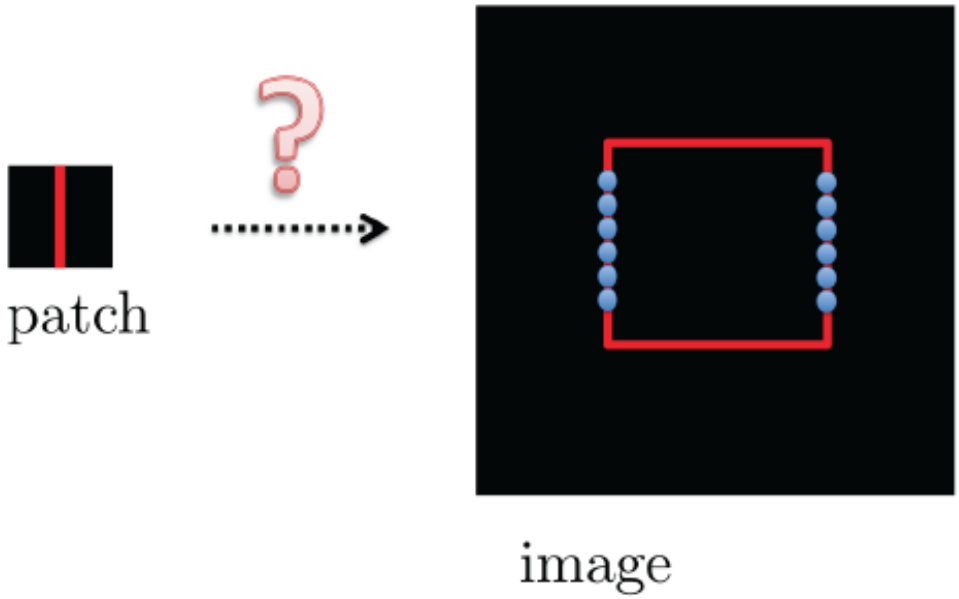
Textureless patches are nearly impossible to localize.

Corners are useful



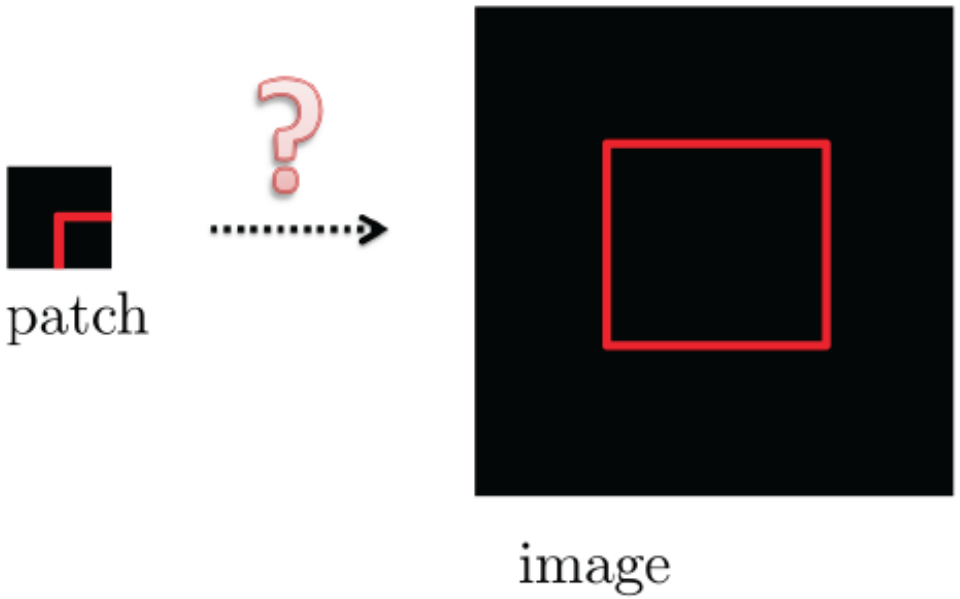
Patches with large contrast are easier to find.

Corners are useful



Patches with large contrast are easier to find, but edges cannot be localized.

Corners are useful

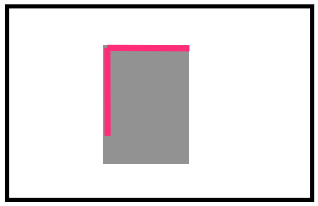


Corners are easier!!

Detection of Corner Features

Need two strong edges:

Example:



Create the following matrix:

$$C = \begin{bmatrix} \sum E_x^2 & \sum E_x E_y \\ \sum E_x E_y & \sum E_y^2 \end{bmatrix}$$

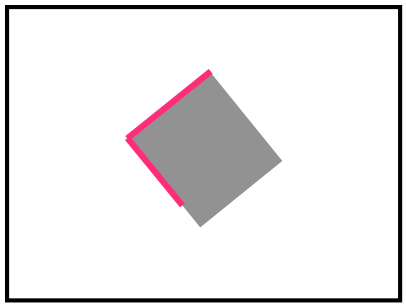
Either E_x or E_y but not both are large in a neighborhood of corner

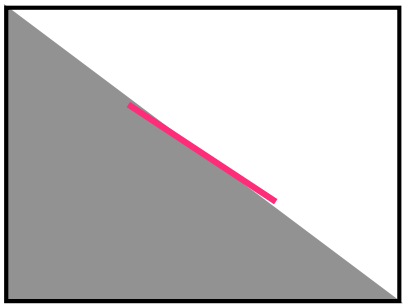
$$\Rightarrow \quad C = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

If $\min(\lambda_1, \lambda_2) > T$
There is a corner!

Detection of Corner Features

What if the corner is not aligned with the image coordinate system?





$$C = \begin{bmatrix} \sum E_x^2 & \sum E_x E_y \\ \sum E_x E_y & \sum E_y^2 \end{bmatrix}$$

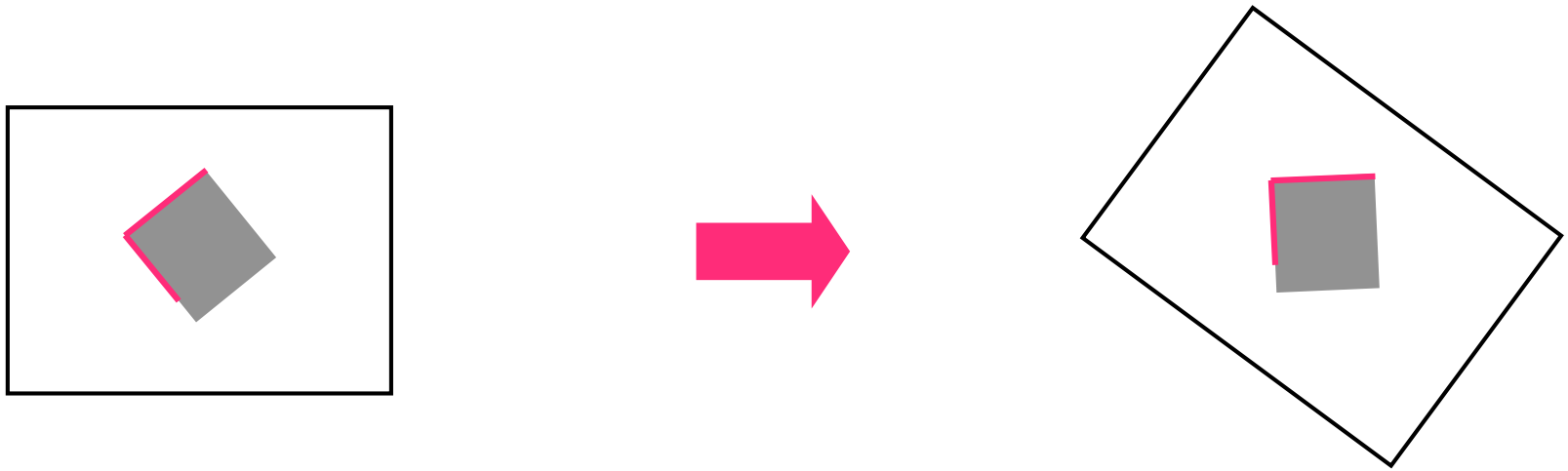
But this is also true for a slanted edge!

Both, E_x or E_y are large in the neighborhood of the corner

Detection of Corner Features

Solution:

"Rotate" the corner to align it with the image coordinate system!



$$C = \begin{bmatrix} \sum E_x^2 & \sum E_x E_y \\ \sum E_x E_y & \sum E_y^2 \end{bmatrix} \qquad C = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

Detection of Corner Features

How do we do this rotation?

Since C is symmetric, it can be diagonalized;

the diagonalization is done by the rotation we need!

Review:Matrix Eigenvalues & Eigenvectors

Let

A be a SQUARE matrix

v be a column vector

λ an scalar

If $A.v = \lambda v$

Assuming the non trivial solution ($v \neq 0$)

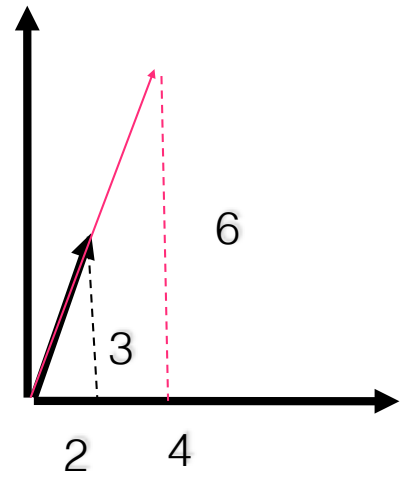
v is an EIGENVECTOR of A

λ is an EIGENVALUE of A

Example:

$$A = \begin{bmatrix} 5 & -2 \\ 6 & -2 \end{bmatrix}$$

$$v = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$



$$Av = \begin{bmatrix} 5 & -2 \\ 6 & -2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix} = 2 \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

v is an eigenvector with eigenvalue 2

Finding Eigenvalues

$$Av = \lambda v$$

$$Av - \lambda v = 0$$

$$(A - \lambda I)v = 0$$

(A - λ I) v = 0 is an HOMOGENEOUS LINEAR system of equations

Finding Eigenvalues

(A - λ I) v = 0 is an HOMOGENEOUS LINEAR system of equations

v is an EV if it is not the TRIVIAL solution v=0

$$det(A - \lambda I) = 0$$

Example:

$$A = \begin{bmatrix} 5 & -2 \\ 6 & -2 \end{bmatrix}$$

$$\det(A - \lambda I) = \det \begin{vmatrix} 5-\lambda & -2 \\ 6 & -2-\lambda \end{vmatrix} = 0$$

$$(5-\lambda)(-2-\lambda) - (6)(-2) = 0$$

$$\lambda^2 - 3\lambda + 2 = 0 \;\Rightarrow \lambda = \frac{3 \pm \sqrt{9-8}}{2}$$

$$\lambda_1 = 1, \lambda_2 = 2$$

Eigenvector for λ = 1

$$\begin{aligned} (5-1)v_{11} - 2v_{12} &= 0 \\ 6v_{11} + (-2-1)v_{12} &= 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} 4v_{11} - 2v_{12} &= 0 \\ 6v_{11} - 3v_{12} &= 0 \end{aligned}$$

$$\Rightarrow 2v_{11} = v_{12} \Rightarrow \quad v_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Eigenvector for λ = 2

$$\begin{aligned} (5-2)v_{21} - 2v_{22} &= 0 \\ 6v_{21} + (-2-2)v_{22} &= 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} 3v_{21} - 2v_{22} &= 0 \\ 6v_{21} - 4v_{22} &= 0 \end{aligned}$$

$$\Rightarrow 3v_{21} = 2v_{22} \quad \Rightarrow \quad v_2 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

Properties:

Trace(A) = $\sum \lambda_i$

$$Trace(A) = Trace\begin{bmatrix} 5 & -2 \\ 6 & -2 \end{bmatrix} = 5 - 2 = 3$$

$$\sum \lambda_i = 1 + 2 = 3$$

Properties:

det(A) = $\Pi\ \lambda_i$

$$\det(A) = \det\begin{bmatrix} 5 & -2 \\ 6 & -2 \end{bmatrix} = 5(-2) - 6(-2) = 2$$

$$\Pi\lambda_i = 1 \cdot 2 = 2$$

Properties:

If there is a NULL eigenvalue, A is not invertible: det(A)=0

A and A^2 have the same eigenvectors.

If λ is and EV of A, λ^2 is an EV of A^2

Symmetric Matrix

$$A = A^T$$

With ev v_i and v_j and eval λ_i λ_j

v_i is an ev:

$$Av_i = \lambda_i v_i$$

Multiply both sides by v_j^T

$$\underbrace{v_j^T A} v_i = \lambda_i v_j^T v_i$$

A is symmetric:

$$v_j^T A = v_j^T A^T$$

v_j is an ev:

$$v_j^T A^T = \lambda_j v_j^T$$

$$\lambda_j v_j^T v_i = \lambda_i v_j^T v_i$$

Symmetric Matrix

$$A = A^T$$

With ev v_i and v_j and eval λ_i λ_j

$$\lambda_j v_j^T v_i = \lambda_i v_j^T v_i$$

$$(\lambda_j - \lambda_i) v_j^T v_i = 0 \quad \Rightarrow \quad v_i \perp v_j$$

The eigenvectors of $A = A^T$ are orthogonal to each other

Matrix Diagonalization:

Let A nxn with:

n DIFFERENT eigenvalues λ_1 λ_2 … λ_n

$v_1, v_2 \ldots v_n$ eigenvectors

$$Av_i = \lambda_i v_i$$

Let P nxn with columns $v_1, v_2, \ldots v_n$:

$$P = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_n \\ | & | & & | \end{bmatrix}$$

Matrix Diagonalization

$$P = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_n \\ | & | & & | \end{bmatrix}$$

$$AP = \begin{bmatrix} | & | & & | \\ Av_1 & Av_2 & \dots & Av_n \\ | & | & & | \end{bmatrix}$$

Matrix Diagonalization

$$AP = \begin{bmatrix} | & | & & | \\ Av_1 & Av_2 & \dots & Av_n \\ | & | & & | \end{bmatrix}$$

$$Av_i = \lambda_i v_i$$

$$AP = \begin{bmatrix} | & | & & | \\ \lambda_1 v_1 & \lambda_2 v_2 & \dots & \lambda_n v_n \\ | & | & & | \end{bmatrix}$$

Matrix Diagonalization

$$AP = \begin{bmatrix} | & | & & | \\ \lambda_1 v_1 & \lambda_2 v_2 & \dots & \lambda_n v_n \\ | & | & & | \end{bmatrix} =$$

$$\underbrace{\begin{bmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_n \\ | & | & & | \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}}_{\Lambda}$$

Matrix Diagonalization

$$AP = P\Lambda$$

$$P^{-1}AP = P^{-1}P\Lambda$$

$$\Lambda = P^{-1}AP$$

The diagonalization matrix P is formed by the eigenvectors of A.

Symmetric Matrix Diagonalization

If $A = A^T$, then the columns of P are orthogonal and

$$P^{-1} = P^T$$

$$\Lambda = P^T A P$$

The matrix P represents a ROTATION

Example:

$$A = \begin{bmatrix} 5 & -2 \\ 6 & -2 \end{bmatrix}$$

$$\lambda_1 = 1, \lambda_2 = 2 \quad v_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad v_2 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \qquad P^{-1} = \begin{bmatrix} -3 & 2 \\ 2 & -1 \end{bmatrix}$$

$$P^{-1}A = \begin{bmatrix} -3 & 2 \\ 4 & -2 \end{bmatrix} \qquad P^{-1}AP = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

Example:

$$A = \begin{bmatrix} 12 & 4 \\ 4 & 12 \end{bmatrix}$$

Symmetric!

$$\lambda_1 = 16, \lambda_2 = 8 \qquad v_1 = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} \; v_2 = \begin{bmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}$$

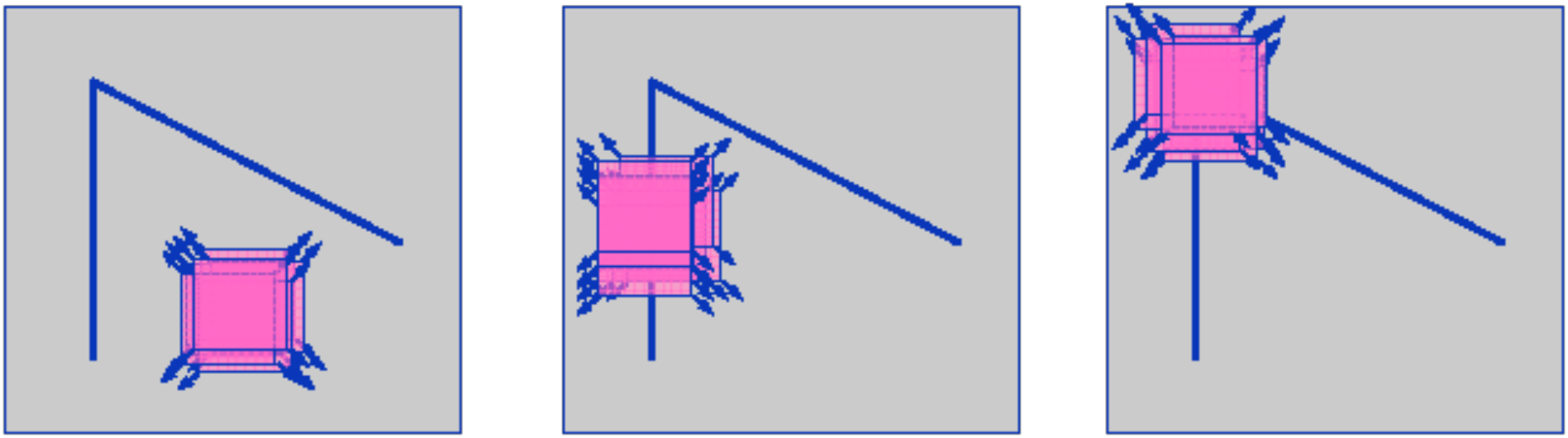
$$P = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \qquad P^{-1} = P^T = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

45° rotation!

$$P^{-1}AP = \begin{bmatrix} 16 & 0 \\ 0 & 8 \end{bmatrix}$$

Harris Corner Detector

The Harris corner detector gives a mathematical approach for determining the amount of changes when we move in all directions a small window.



Harris Corner Detector

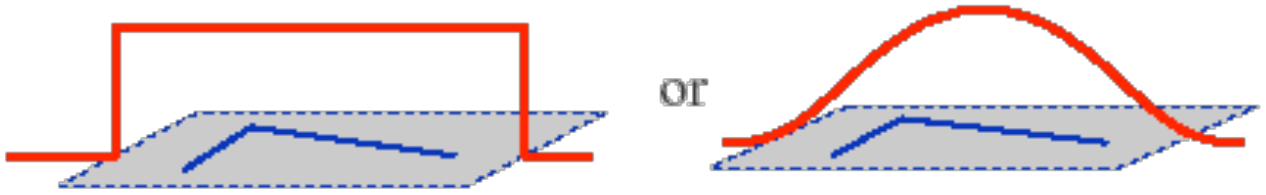
Change of intensity for the shift [u,v]:

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2$$

Window Function

Change In Intensity

Window w(x,y):



1 in window, 0 outside

Gaussian

First Order Approximation of f(x,y):

$$f(x+u, y+v) = f(x,y) + uf_x(x,y) + vf_y(x,y) + \ldots$$

Harris Corner Detector

$$\sum_{x,y}[I(x+u,y+v)-I(x,y)]^2 \approx$$

$$\sum_{x,y}[I(x,y)+uI_x(x,y)+vI_y(x,y)-I(x,y)]^2 =$$

$$\sum_{x,y} u^2 I_x^2 + 2uvI_xI_y + v^2 Iy^2$$

Harris Corner Detector

$$\sum_{x,y} u^2 I_x^2 + 2uvI_xI_y + v^2 Iy^2$$

Rewrite as a Matrix:

$$\sum_{x,y} u^2 I_x^2 + 2uvI_xI_y + v^2 Iy^2 =$$

$$\sum_{x,y} \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} =$$

$$\begin{bmatrix} u & v \end{bmatrix} \left(\sum_{x,y} \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

Harris Detector

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2$$

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

M is computed from the gradient components

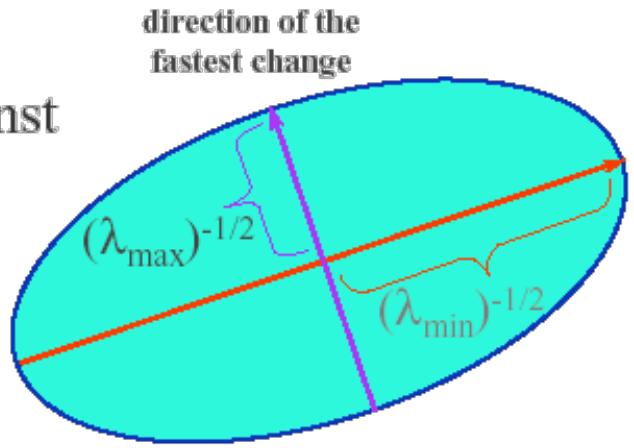
Harris Detector

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

Intensity change in shifting window eigenvalue analysis:

λ_1, λ_2 eigenvalues of M

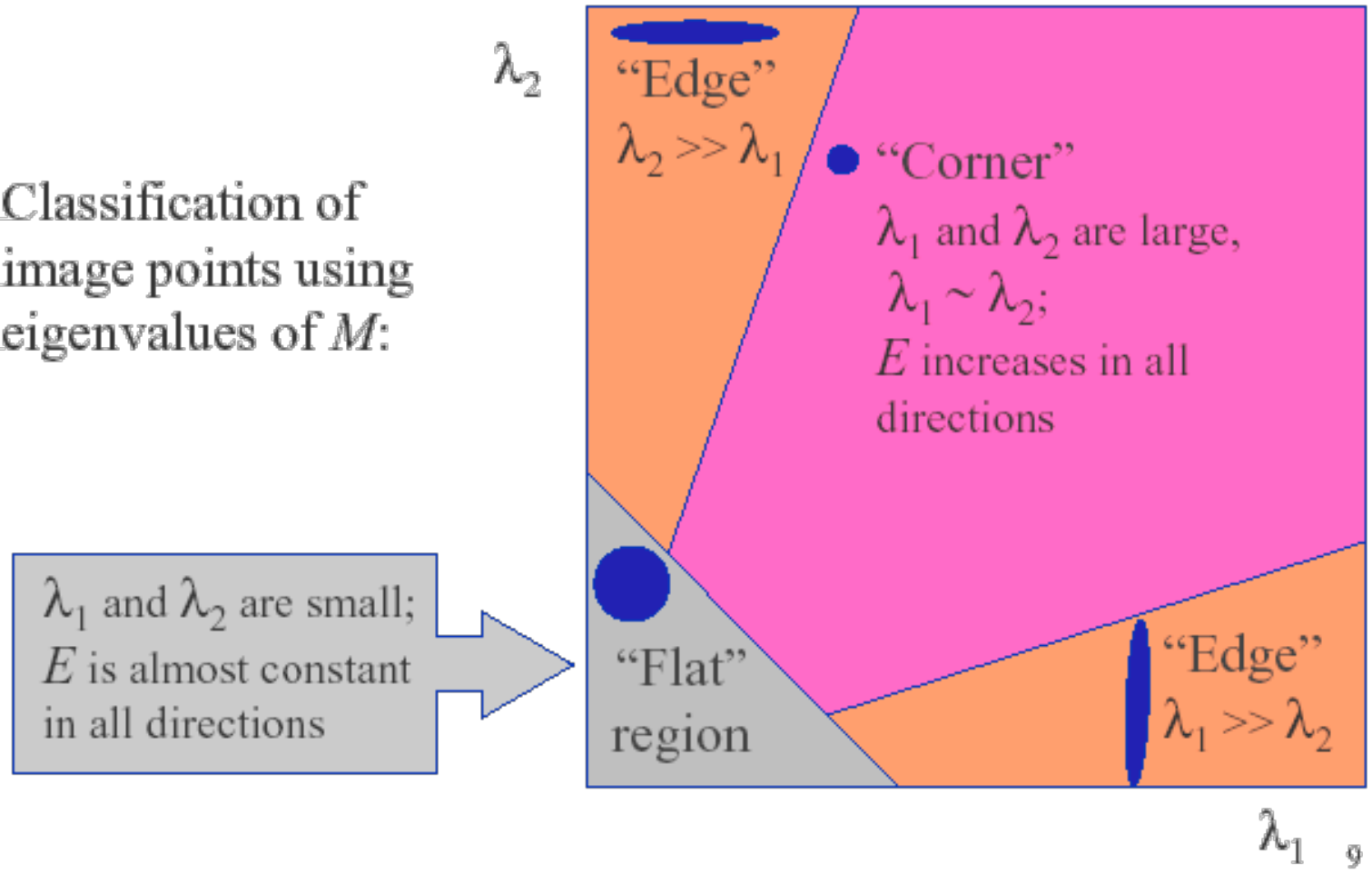
Ellipse $E(u,v) = \text{const}$



8

Classification via Eigenvalues

Classification of image points using eigenvalues of M:



Corner Response Measure

Measure of corner response:

$$R = \det M - k\left(\text{trace}\, M\right)^2$$

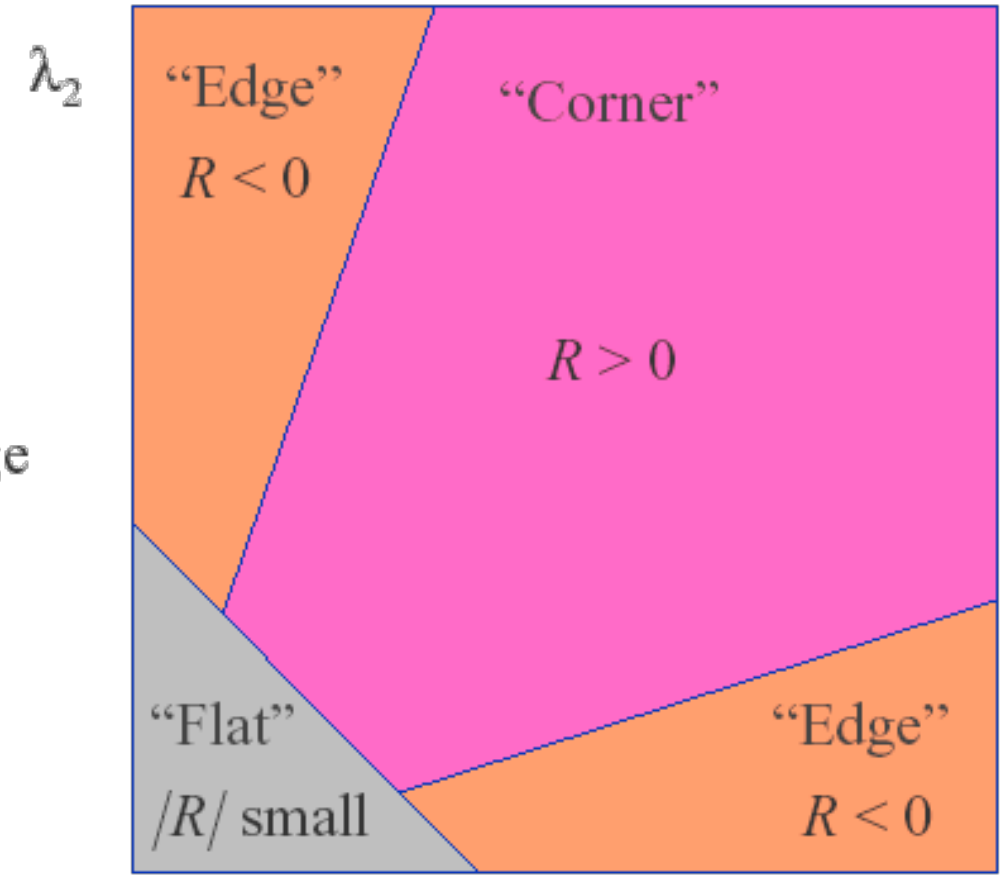
$$\det M = \lambda_1 \lambda_2$$
$$\text{trace}\, M = \lambda_1 + \lambda_2$$

(k is an empirically determined constant; $k = 0.04$ - 0.06)

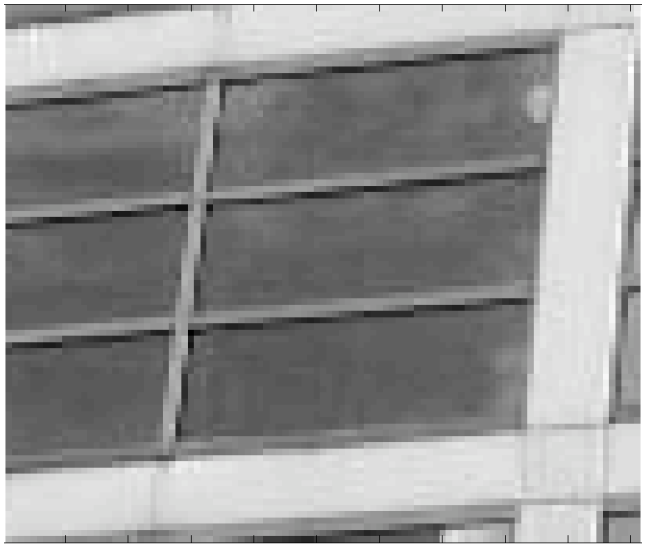
Corner Response Measure

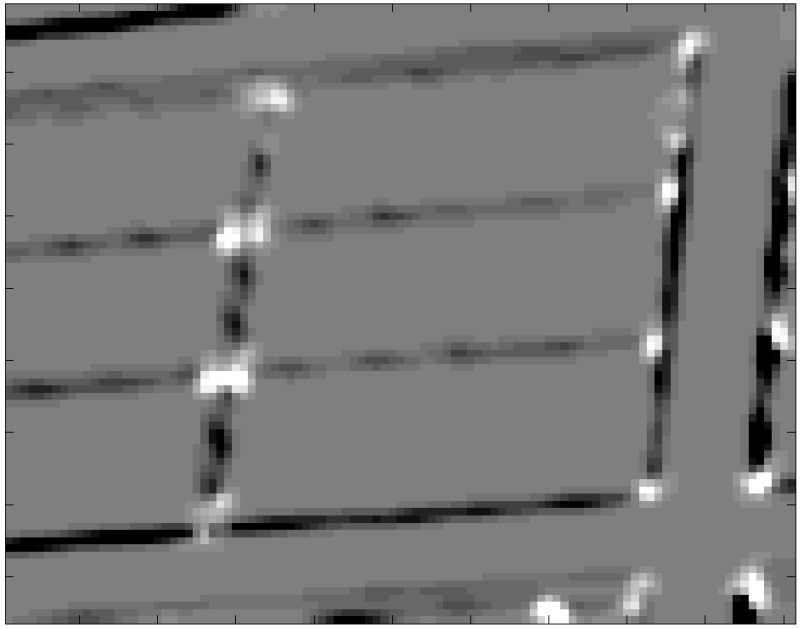
$R = \det M - k \, \text{trace}^2(M)$

- R depends only on eigenvalues of M
- R is large for a corner
- R is negative with large magnitude for an edge
- $|R|$ is small for a flat region



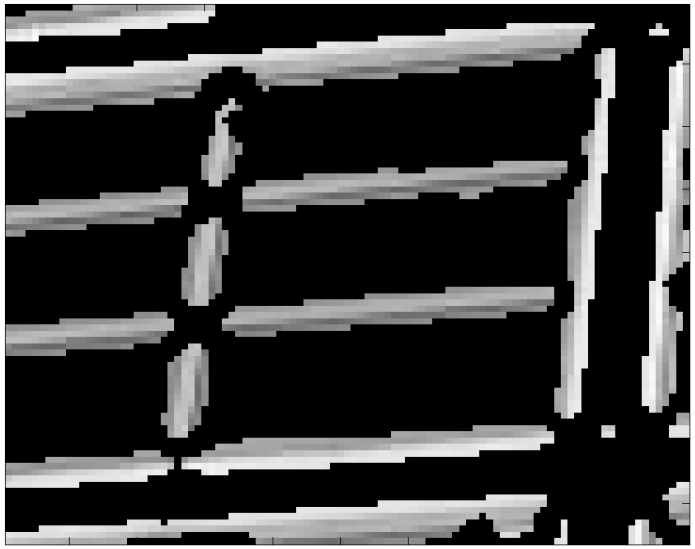
Corner Response Example





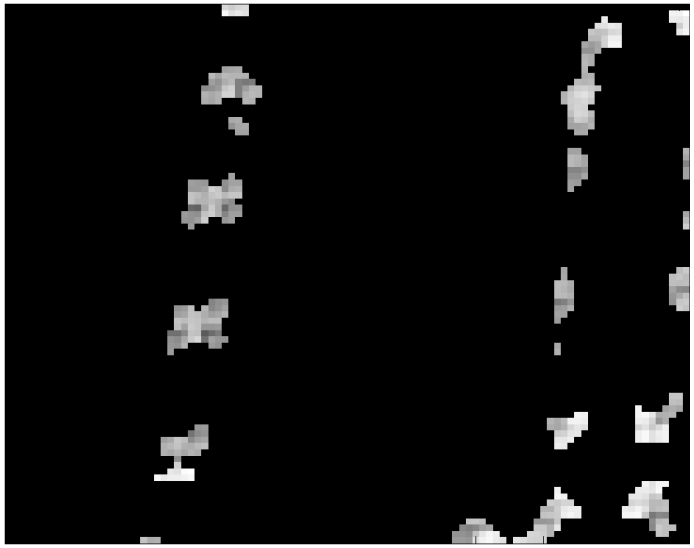
R score; Gradient computed with Sobel mask
Window Gaussian, sigma=1

Corner Response Example: Edges



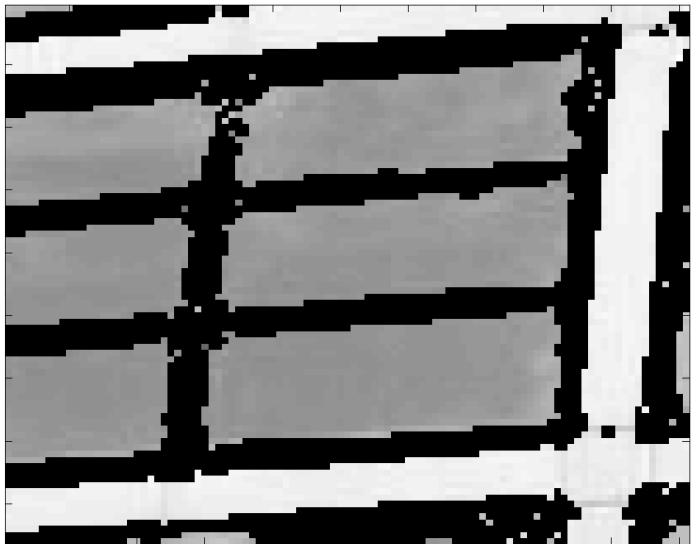
R < - 10000

Corner Response Example: Corners



R > 10000

Corner Response Example:



-10000 < R < 10000

Neither edges nor corners

Harris Corner Detection Algorithm

Compute the Image Gradient

$I_x = G_{x,s} * I$ and $I_y = G_{y,s} * I$

Compute products of derivatives at each pixel

$I^2_x = I_x \cdot I_x$ and $I^2_y = I_y \cdot I_y$ and $I_{xy} = I_x \cdot I_y$

Compute the sums of the products at each pixel using a window averaging:

$S^2_x = G_{s'} * I^2_x$ $S^2_y = G_{s'} * I^2_y$ $S_{xy} = G_{s'} * I_{xy}$

Define the Matrix at each pixel $M = [S^2_x \; S_{xy} \; ; \; S_{xy} \; S^2_y]$

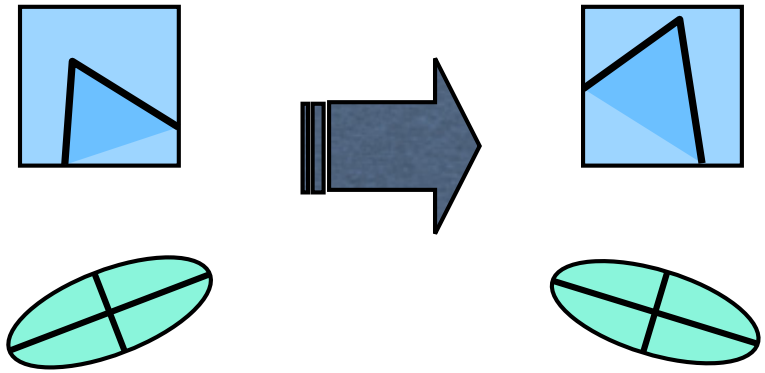
Compute the response $R = \det M - k \, \text{trace}(M)^2$

Threshold R

Compute Nonmax suppression

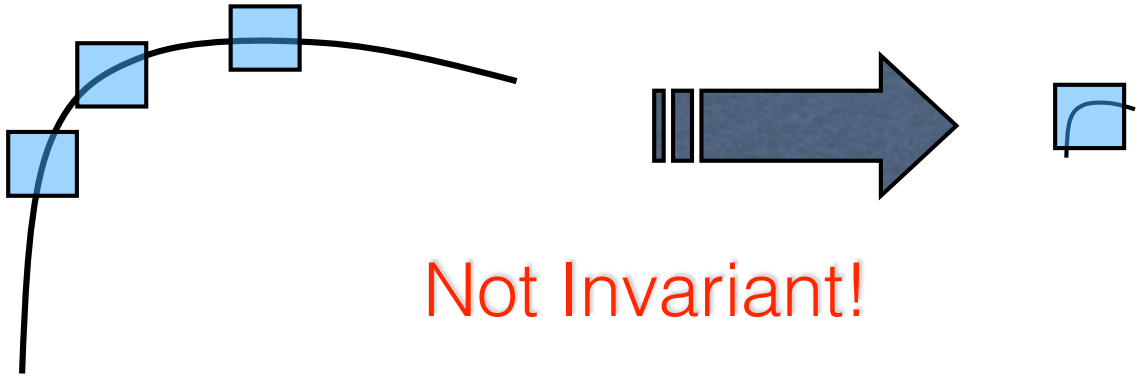
Invariance Properties

Rotation



Eigenvalues do not change -> Invariant

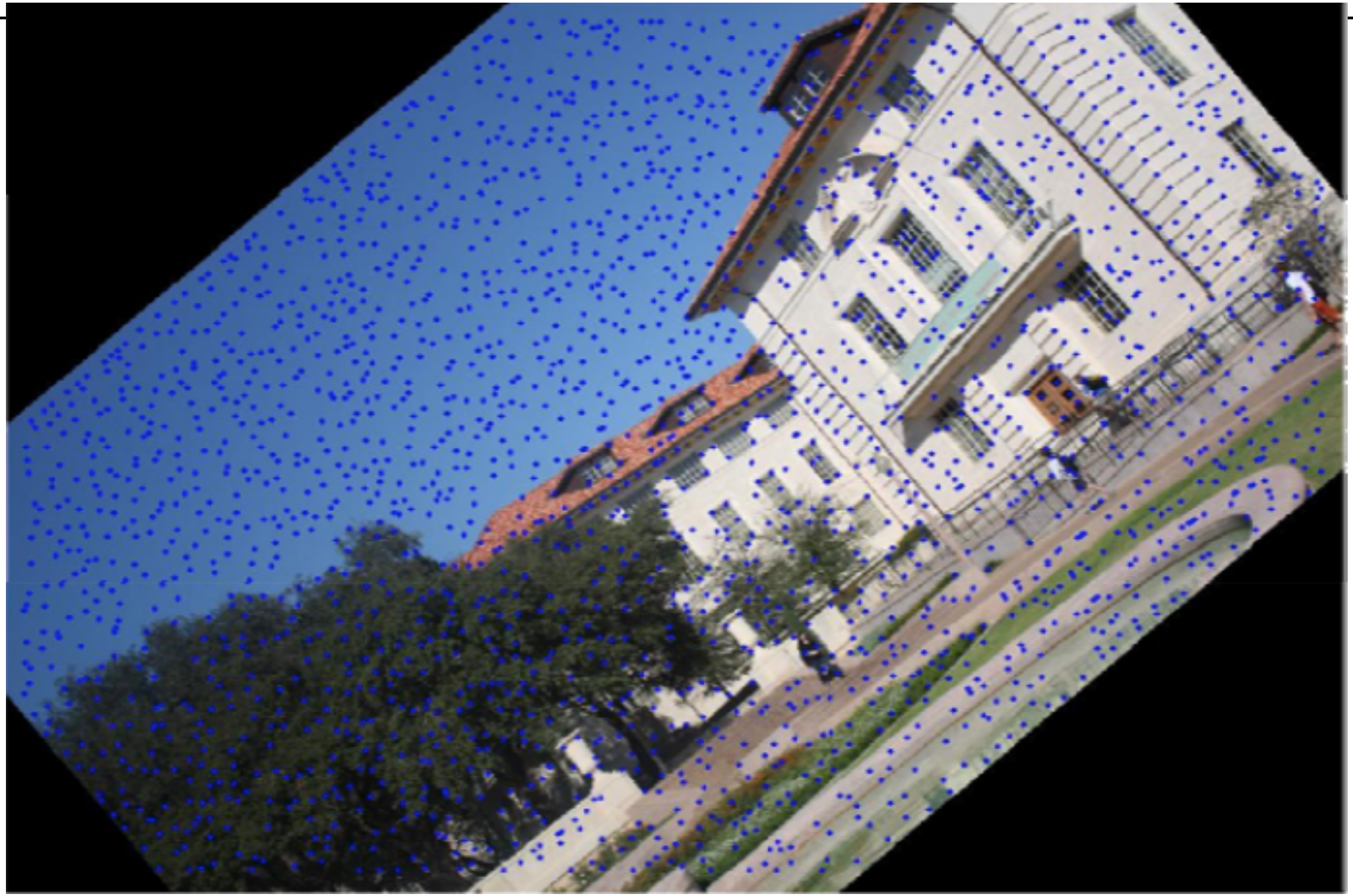
Scale



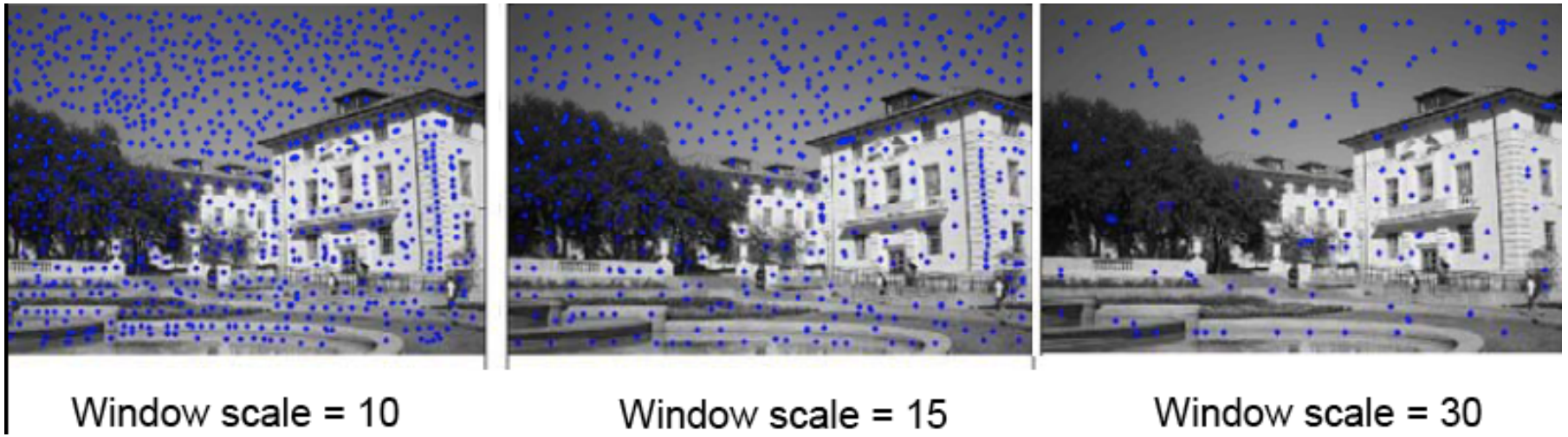
Not Invariant!

Example

Example

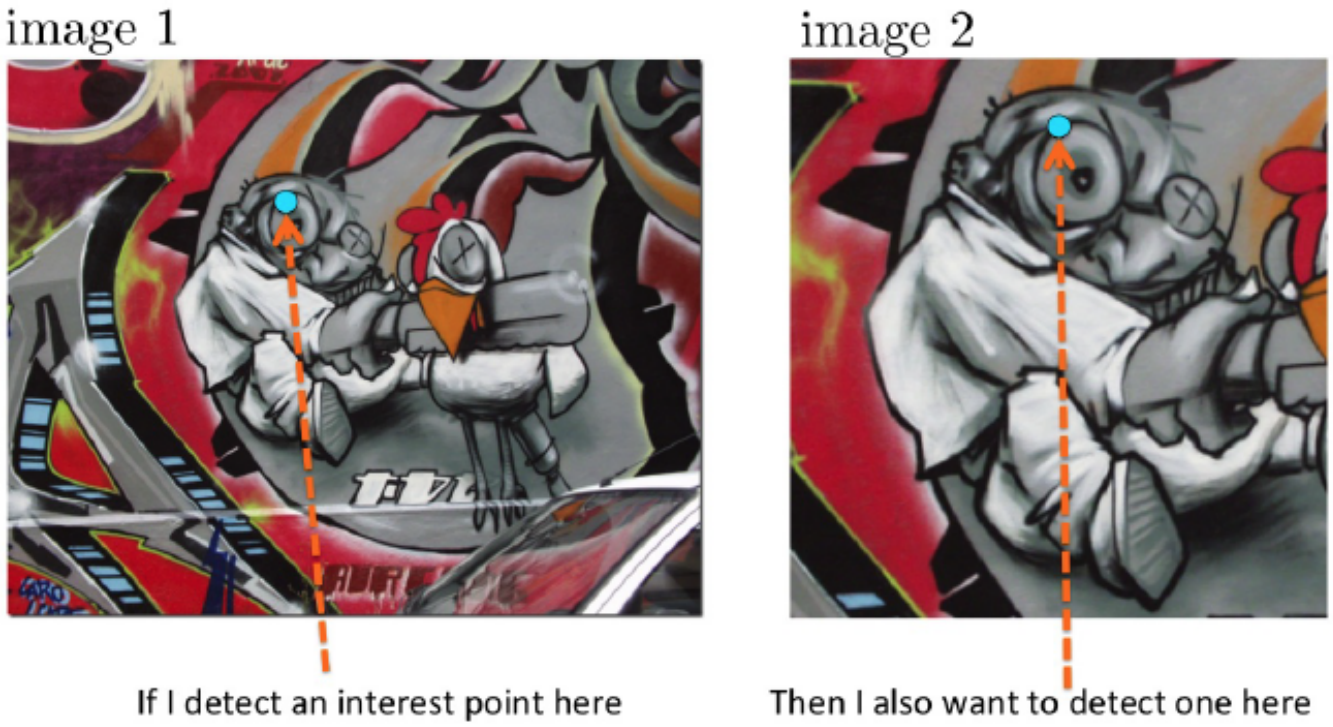


Example



Harris corner detector finds local maxima at different set of points, depending on the scale of the window we sum over.

Scale Invariant Interest Points



Scale Invariant Interest Points

How can we independently select interest points in each image, such that the detections are repeatable across different scales?

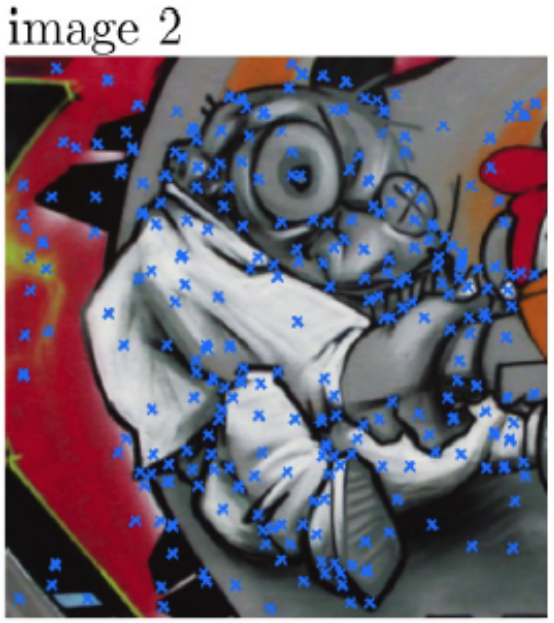
image 1

image 2

Scale Invariant Interest Points

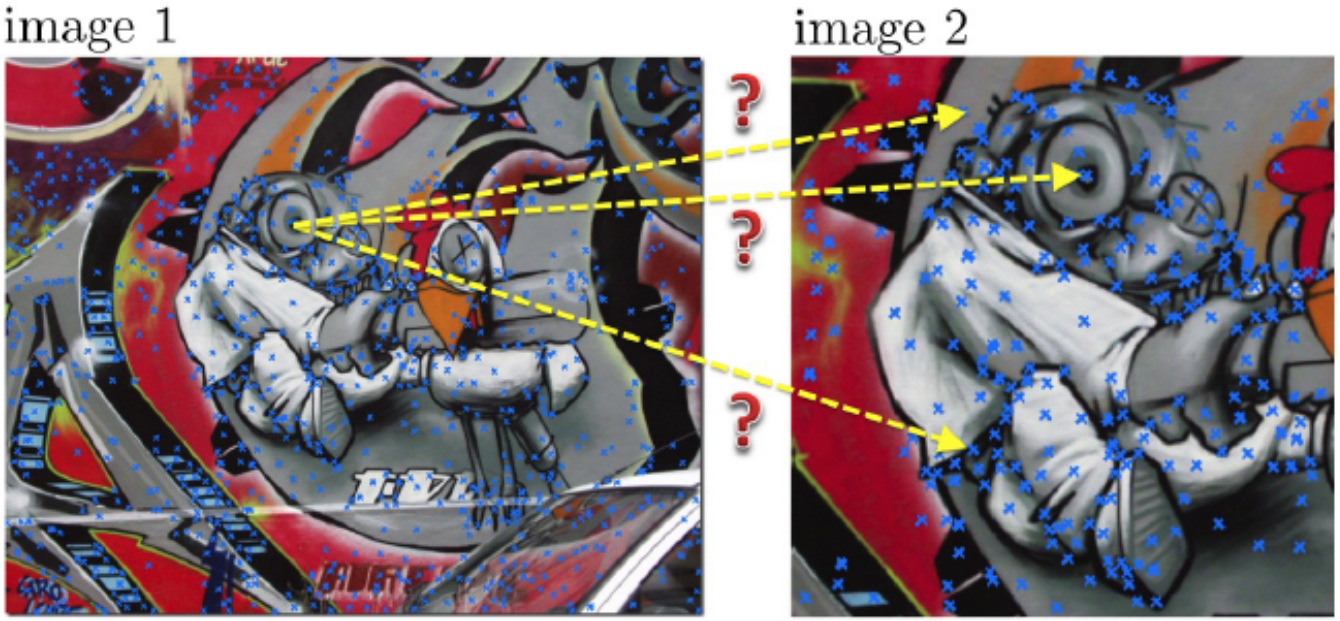
image 1

Scale Invariant Interest Points



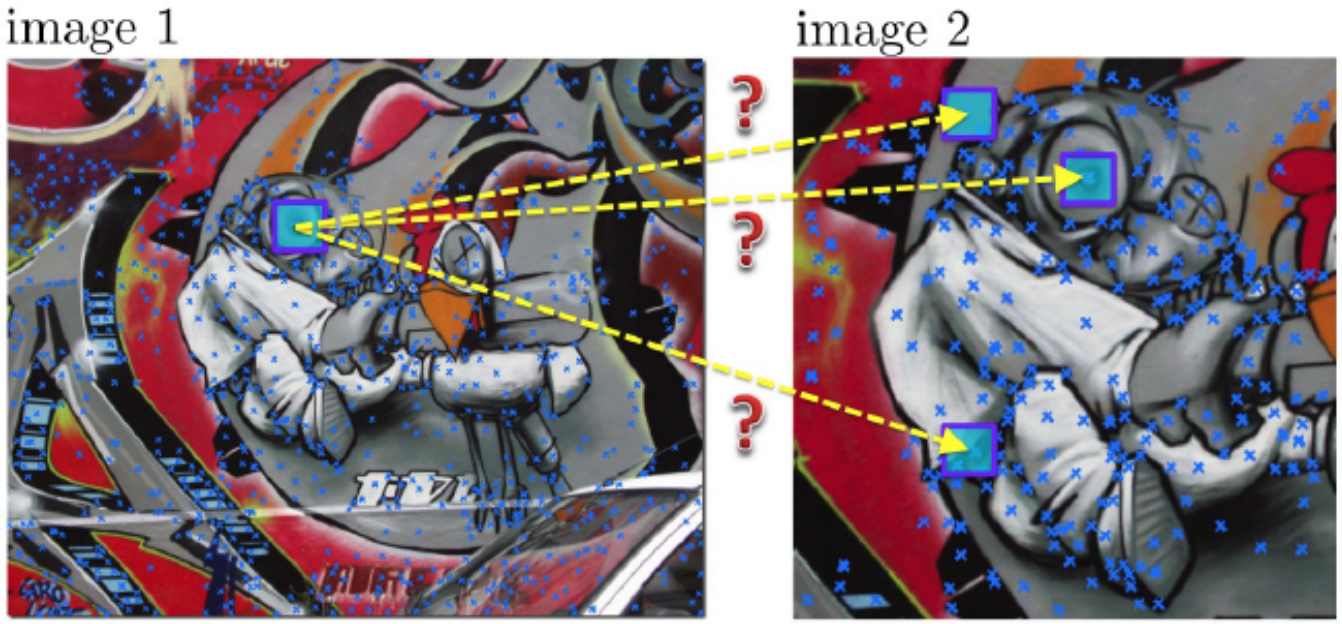
Scale Invariant Interest Points

Our goal is to match objects in different images



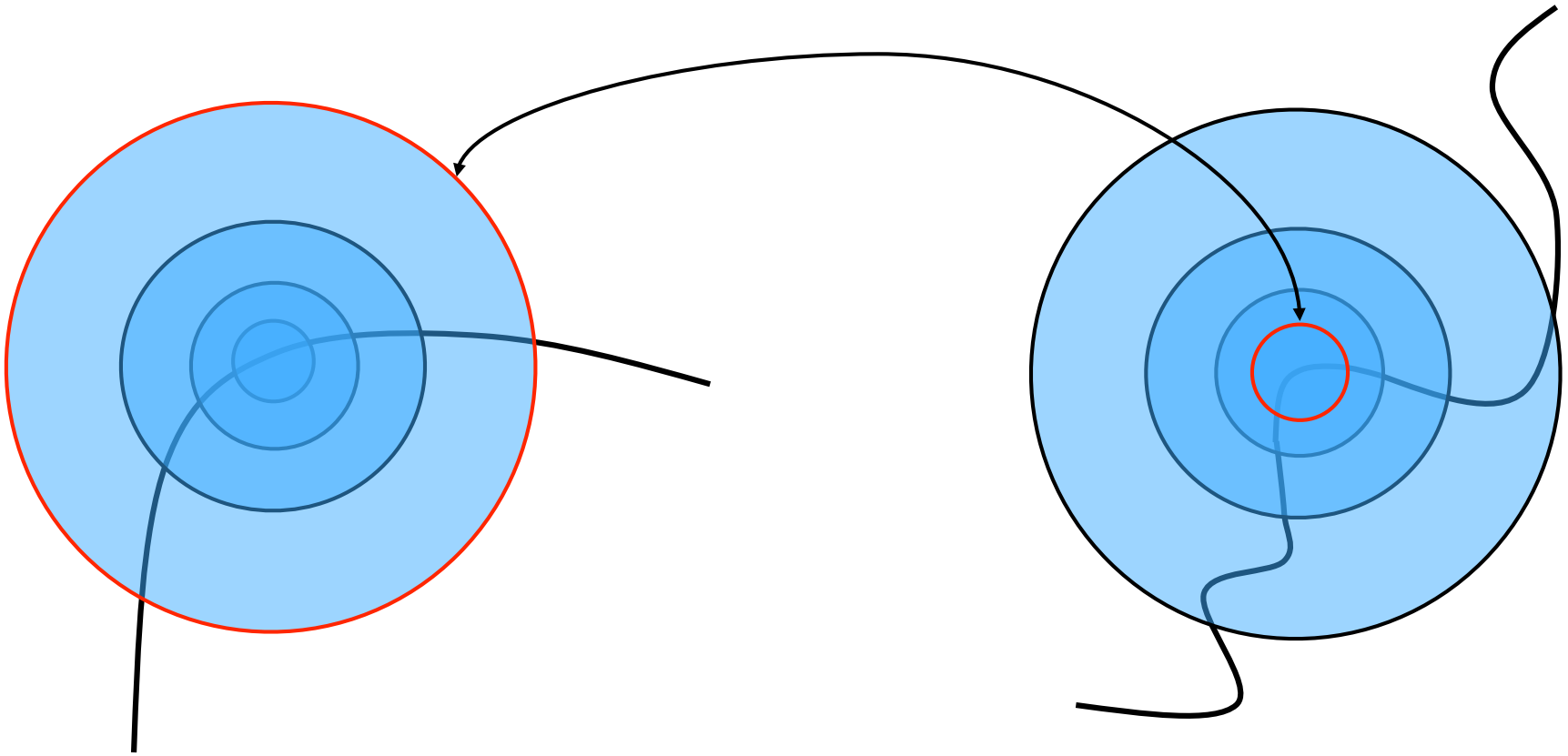
Scale Invariant Interest Points

Our goal is to match objects in different images



- Find interest points in each image
- Form a vector description of each point
- Compare the vectors

Scale Invariant Detection



Intuition: Find scale that gives local maxima of some function f in both, position and scale.

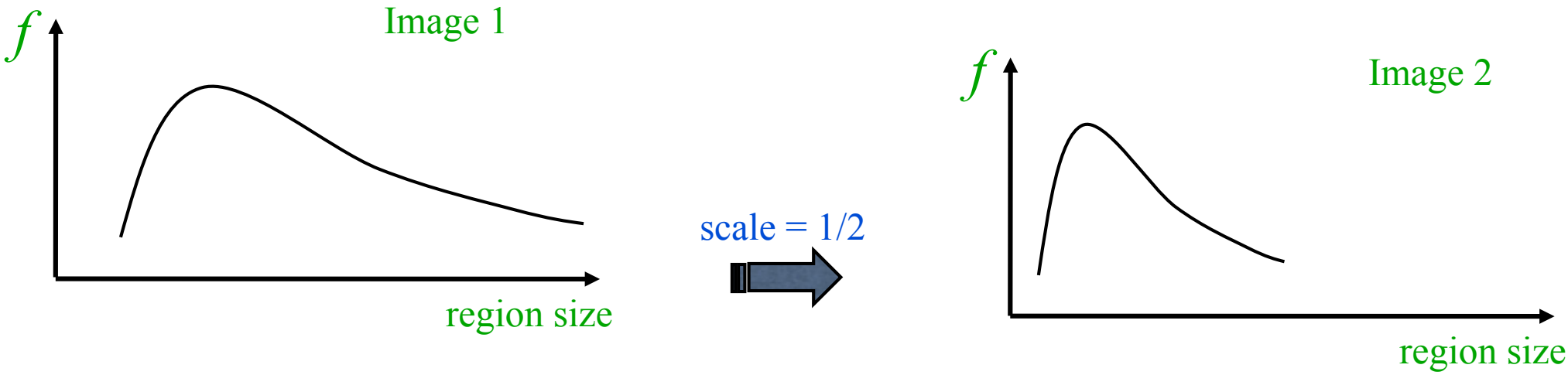
Scale Invariant Detection

Design a function on the region (circle), which is "scale invariant"

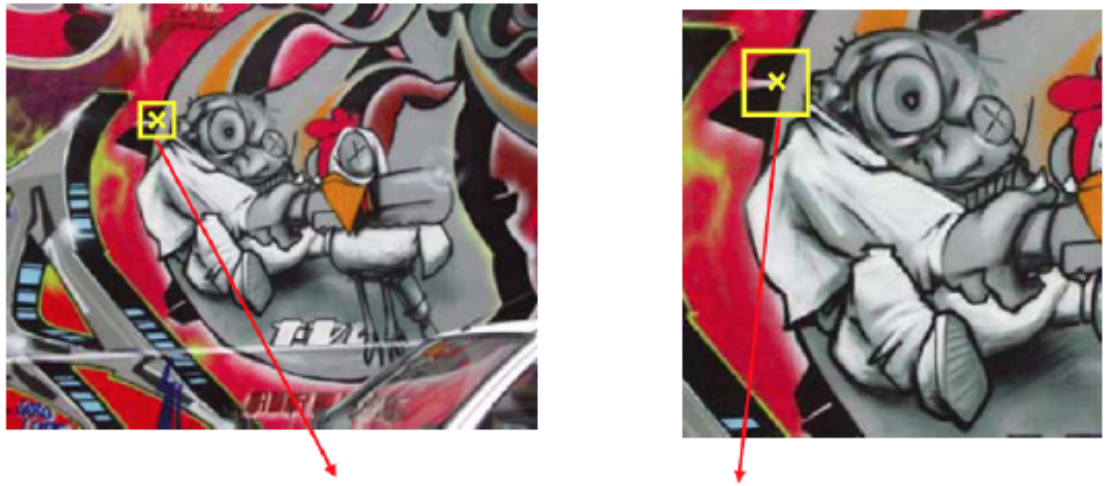
(the same for corresponding regions, even if they are at different scales)

- For a point in one image, we can consider it as a function of region size (circle radius)
- Find the local maxima (scale invariant)

Example: average intensity. For corresponding regions (even of different sizes) it will be the same.



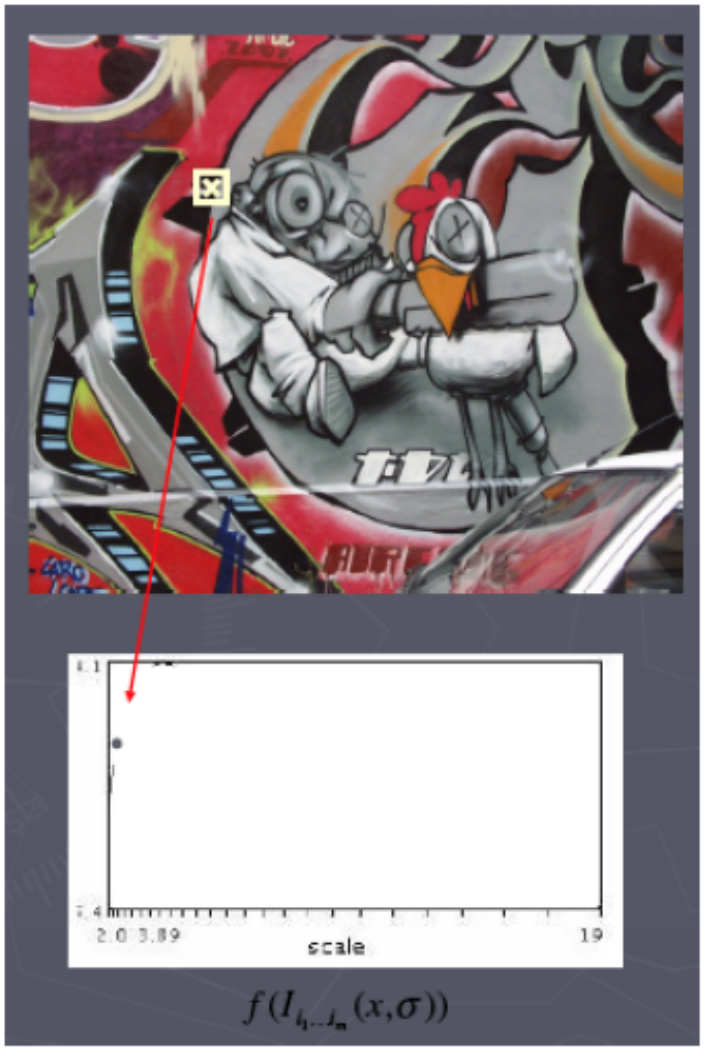
Scale at Local Maxima



$$f(I_{i_1 \ldots i_m}(x,\sigma)) = f(I_{i_1 \ldots i_m}(x',\sigma'))$$

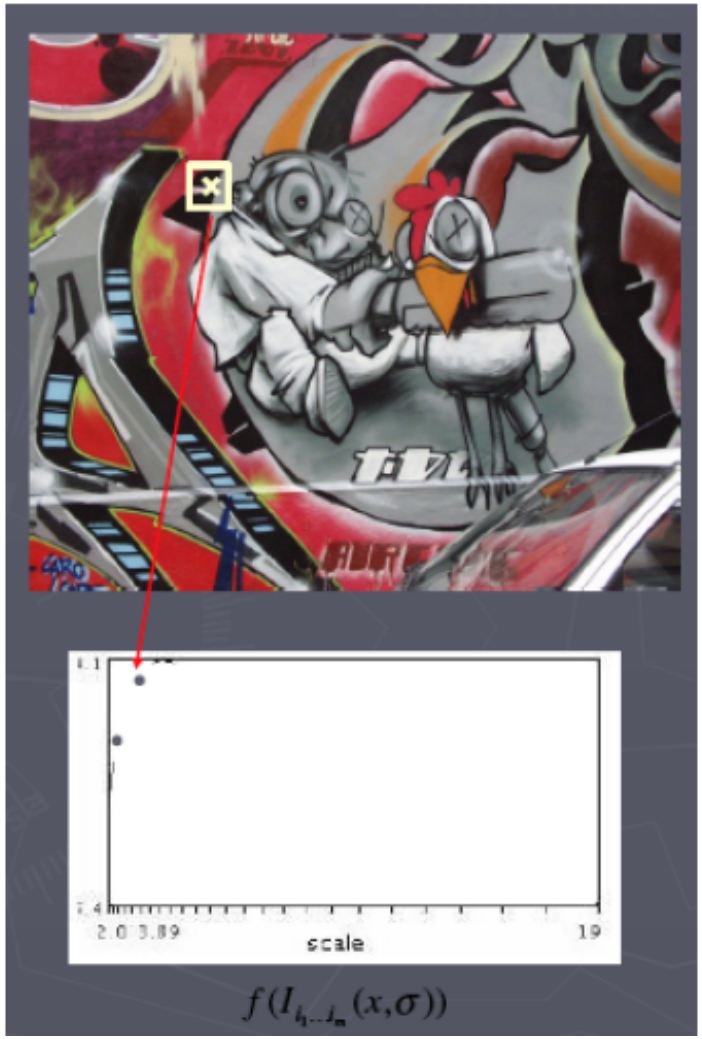
Automatic Scale Selection

Scale signature: Function response for increasing scale



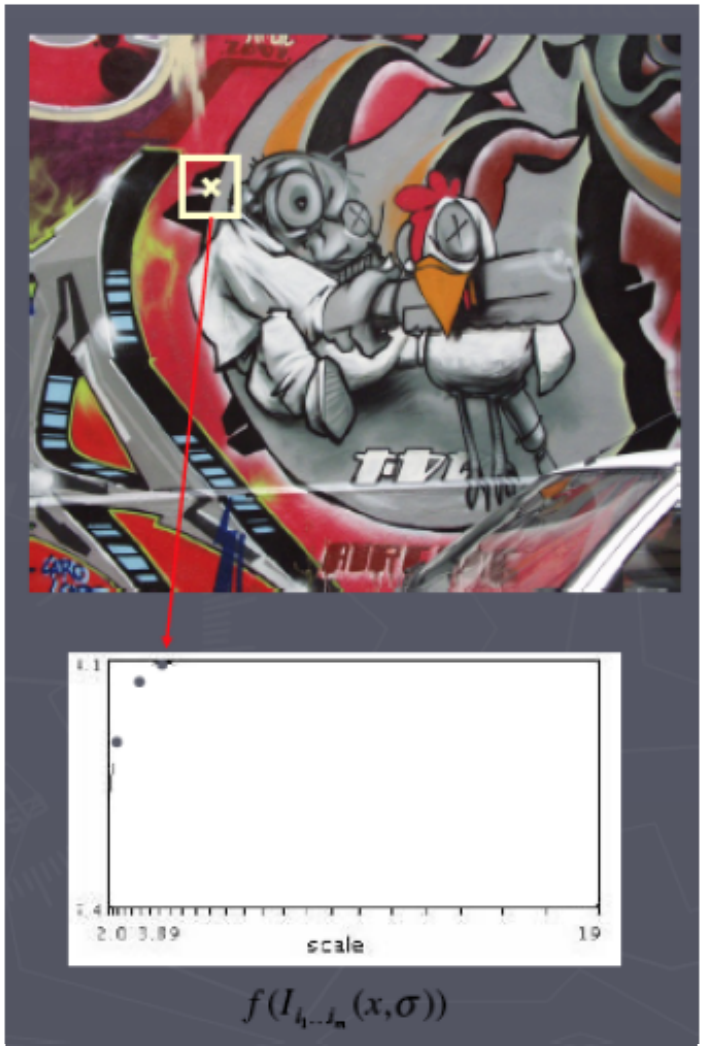
Automatic Scale Selection

Scale signature: Function response for increasing scale



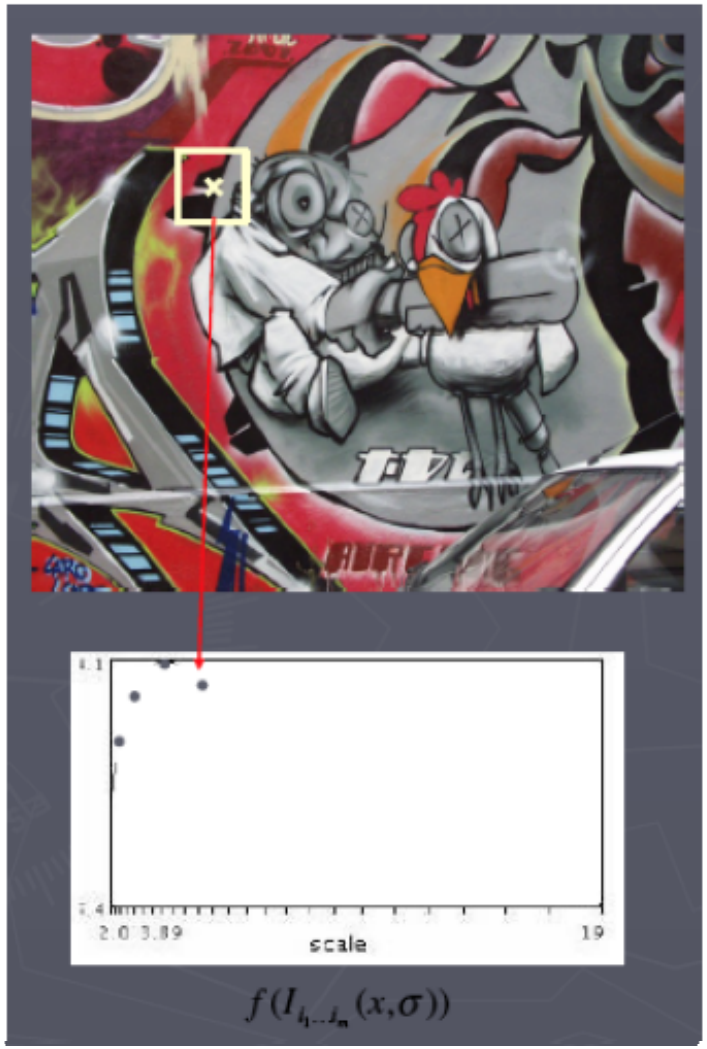
Automatic Scale Selection

Scale signature: Function response for increasing scale



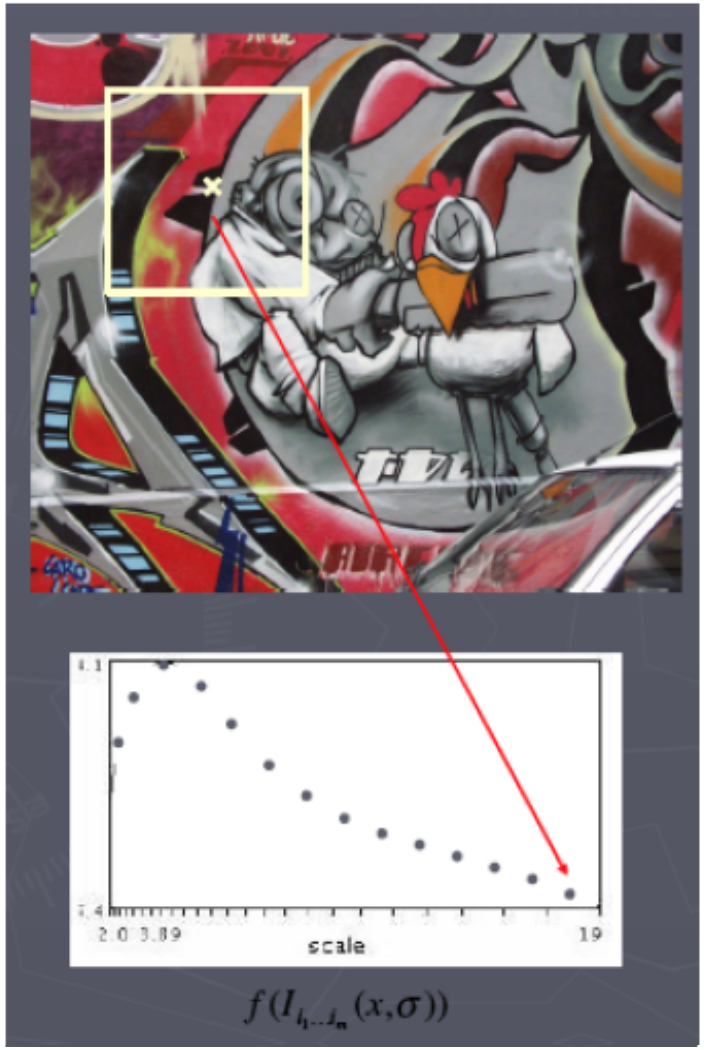
Automatic Scale Selection

Scale signature: Function response for increasing scale



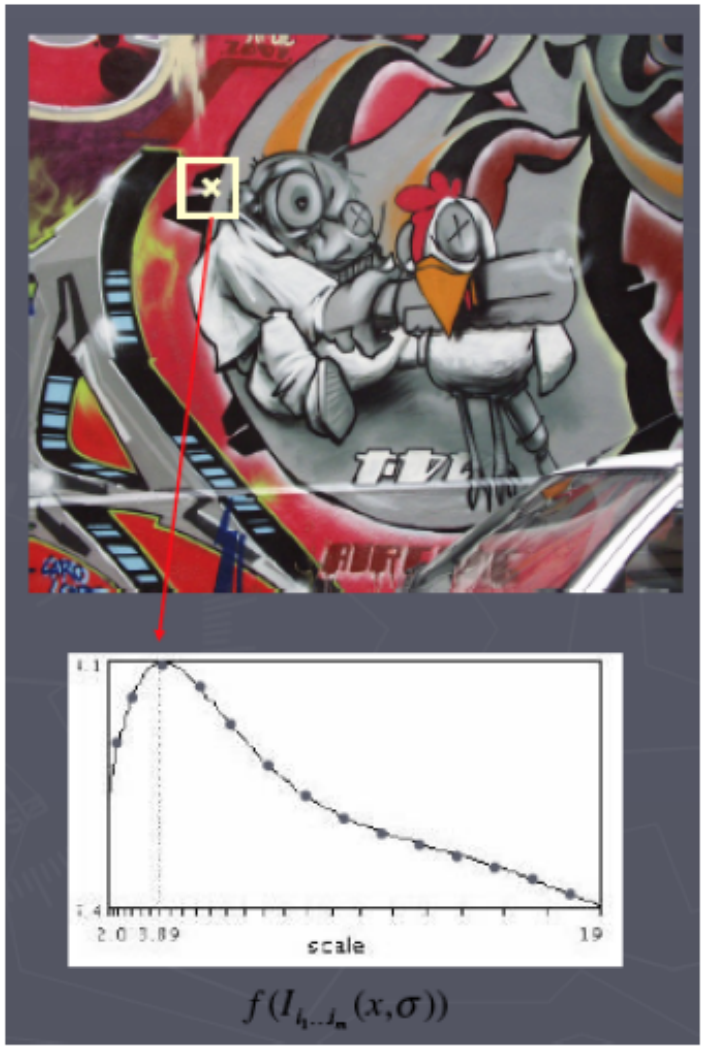
Automatic Scale Selection

Scale signature: Function response for increasing scale



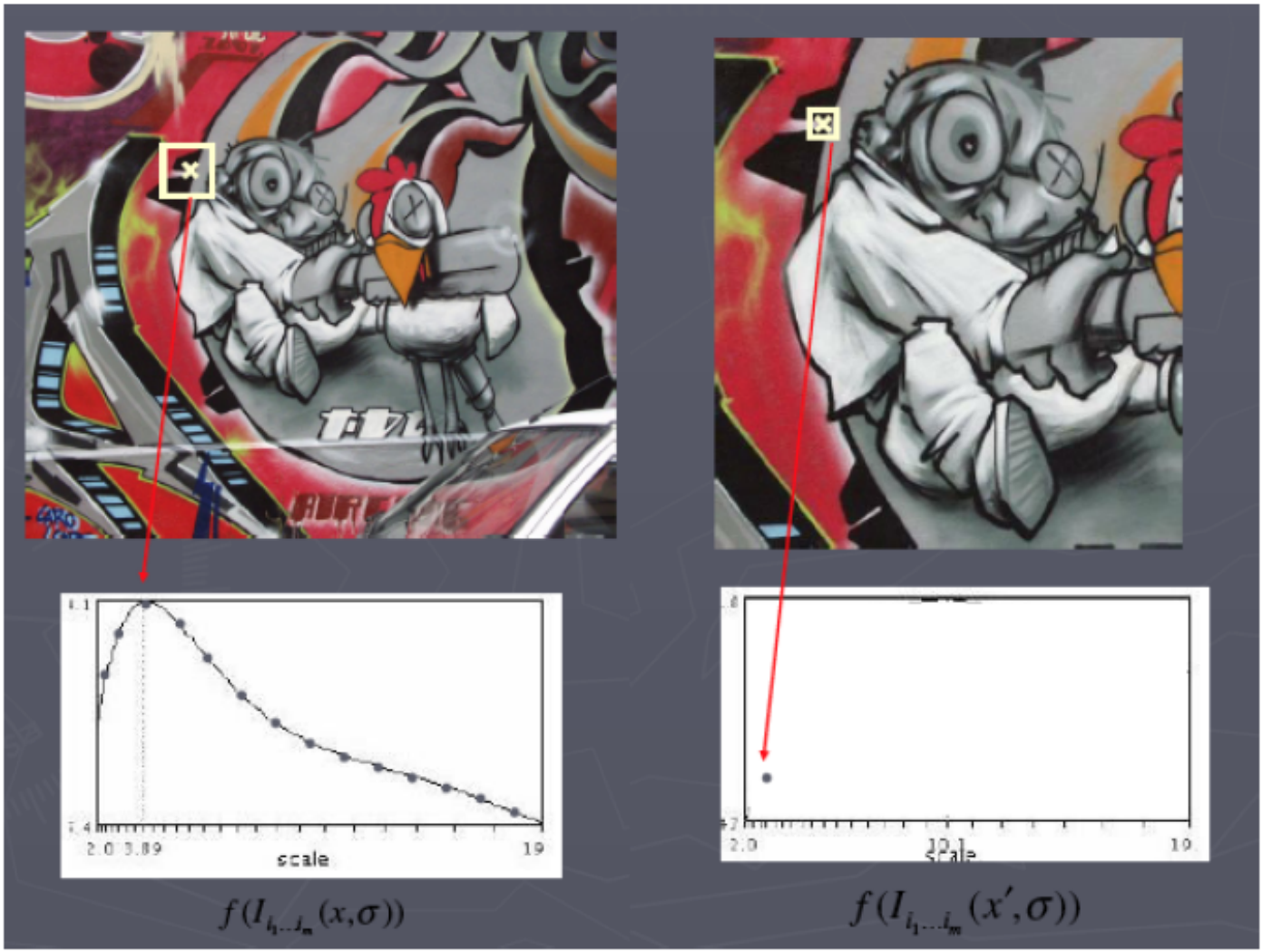
Automatic Scale Selection

Scale signature: Function response for increasing scale



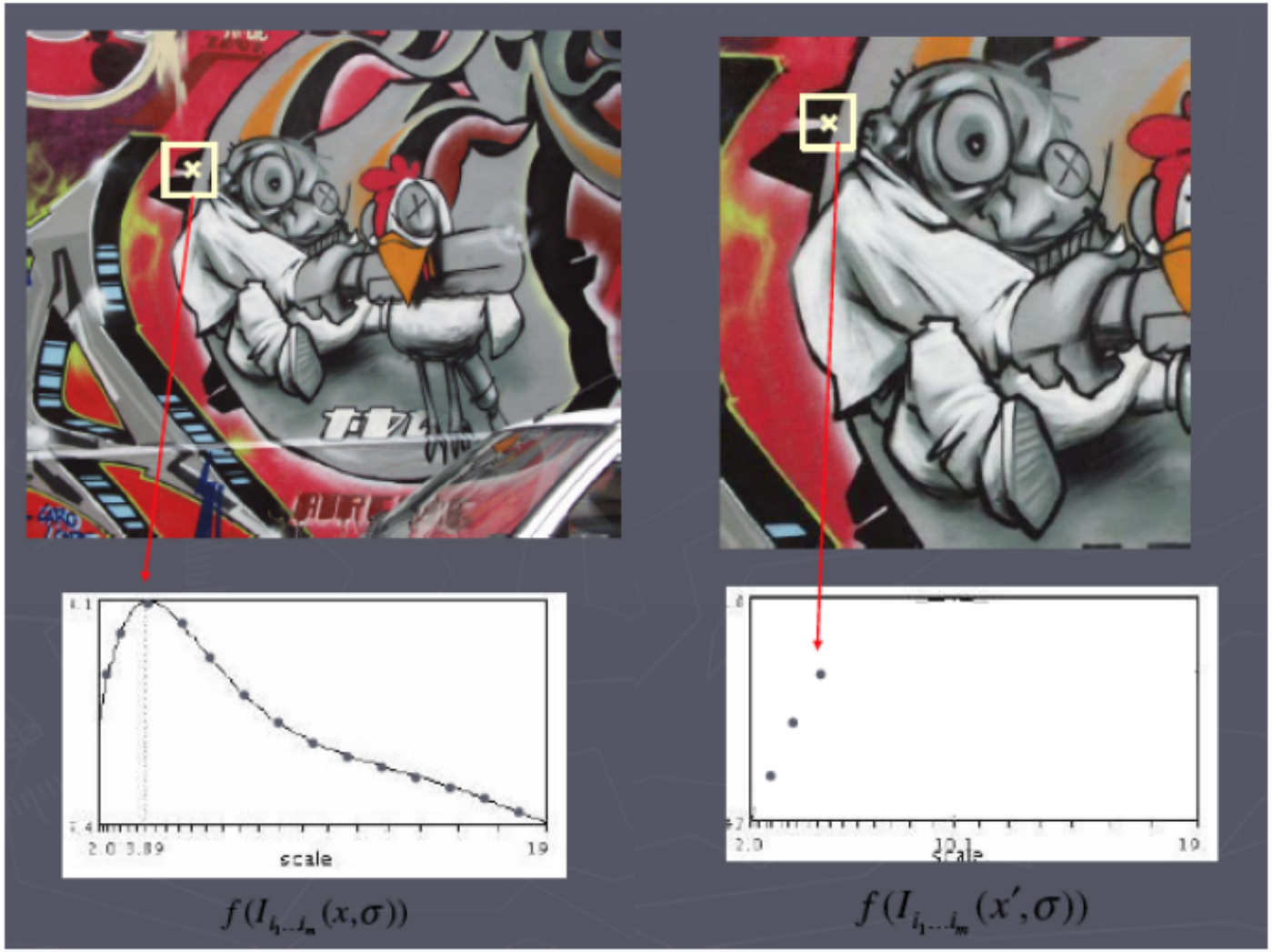
Automatic Scale Selection

Scale signature: Function response for increasing scale



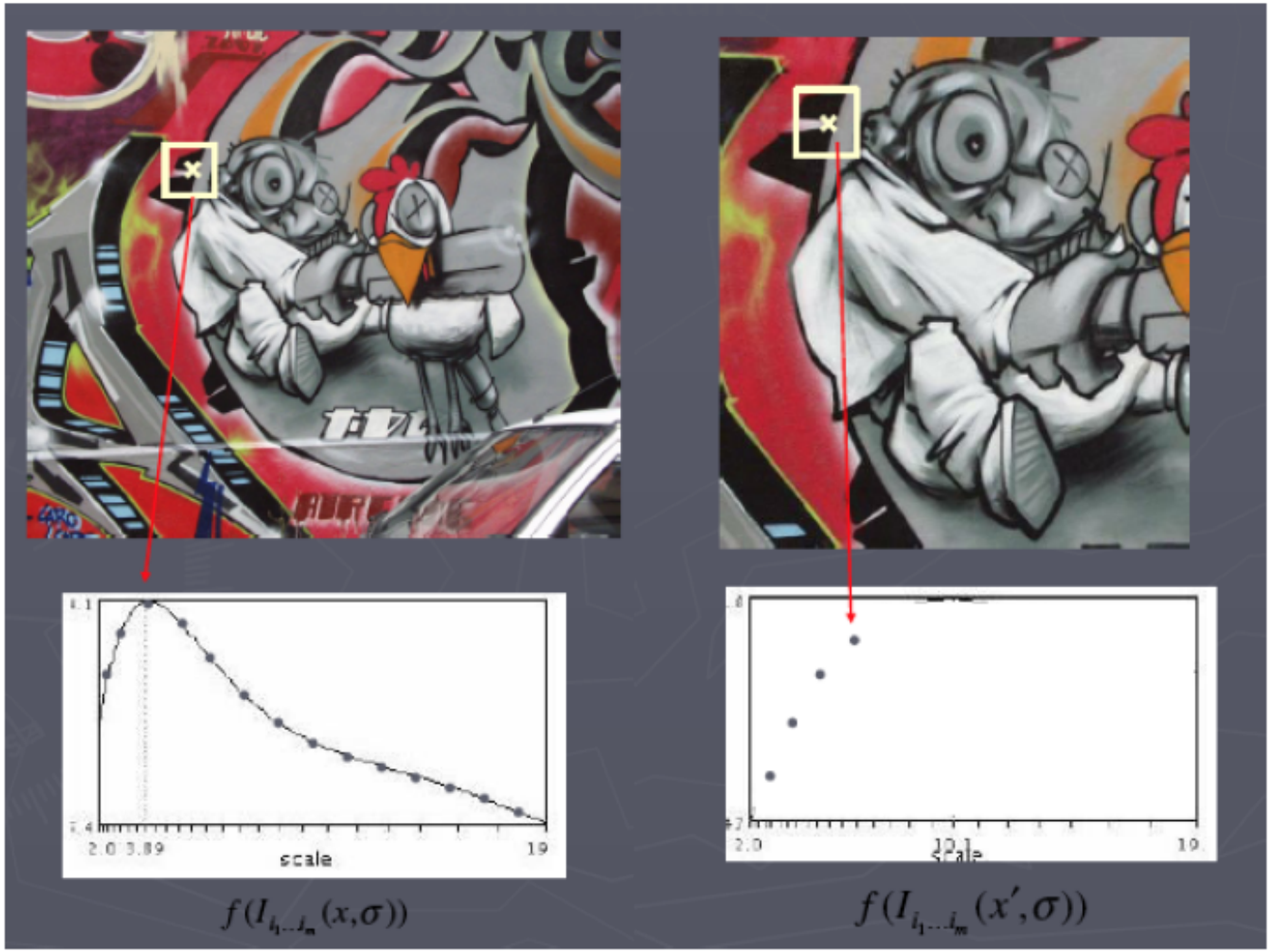
Automatic Scale Selection

Scale signature: Function response for increasing scale



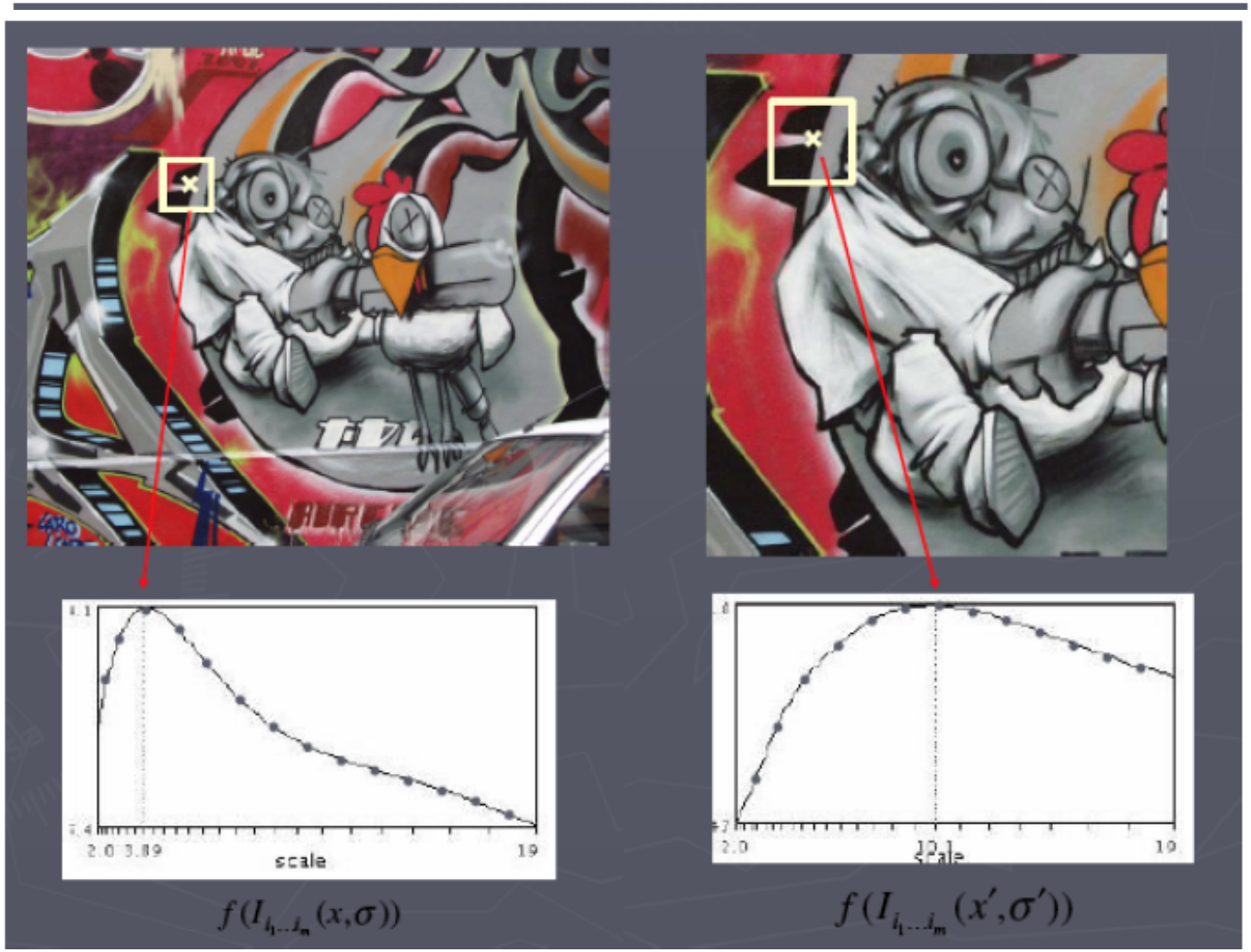
Automatic Scale Selection

Scale signature: Function response for increasing scale



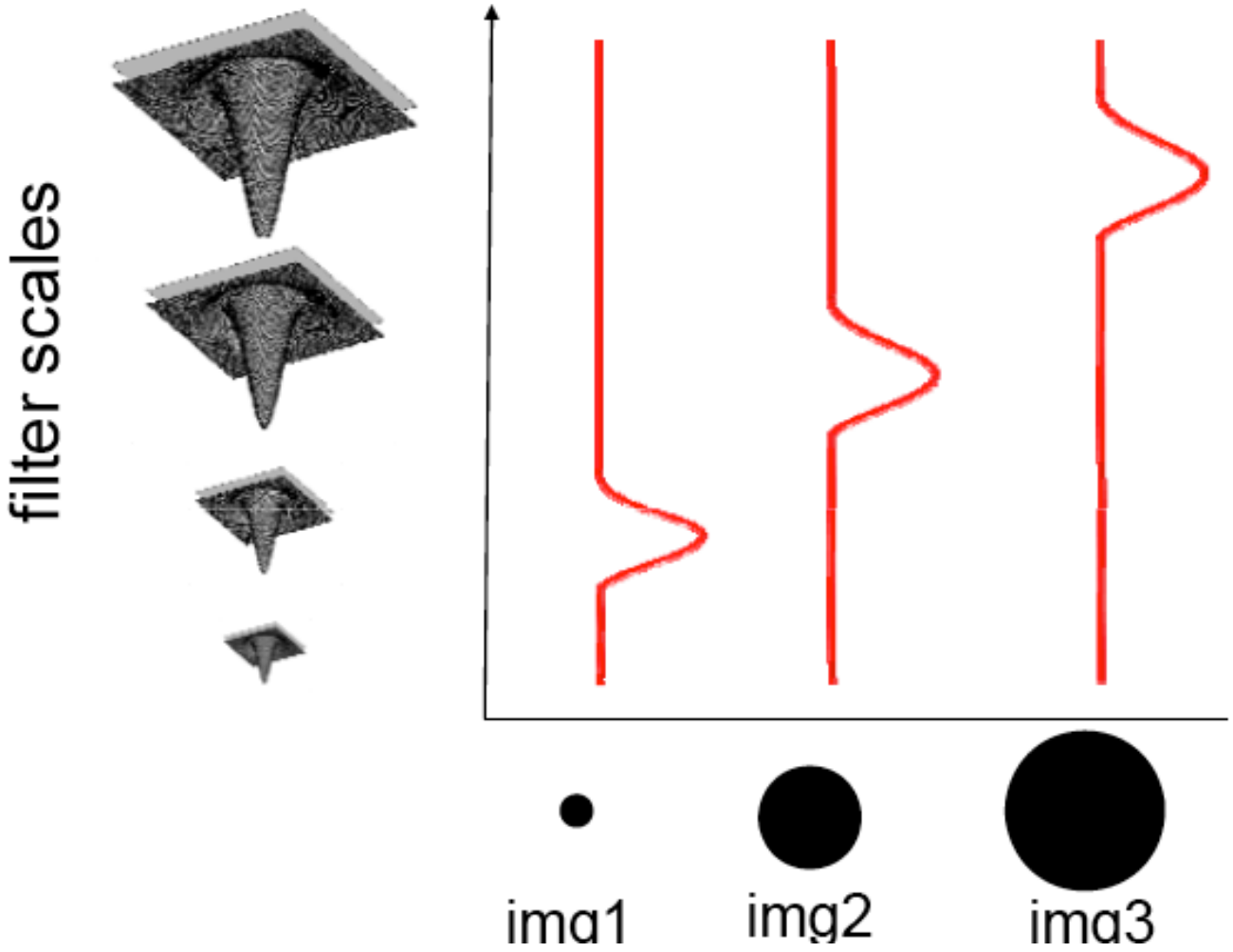
Automatic Scale Selection

Scale signature: Function response for increasing scale

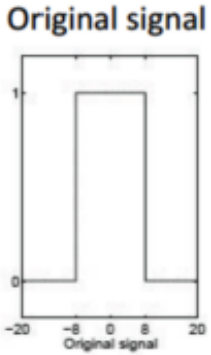


What can be a "signature" function f?

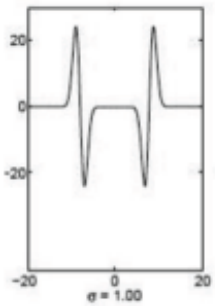
Laplacian-of-Gaussian = "blob" detector $\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$



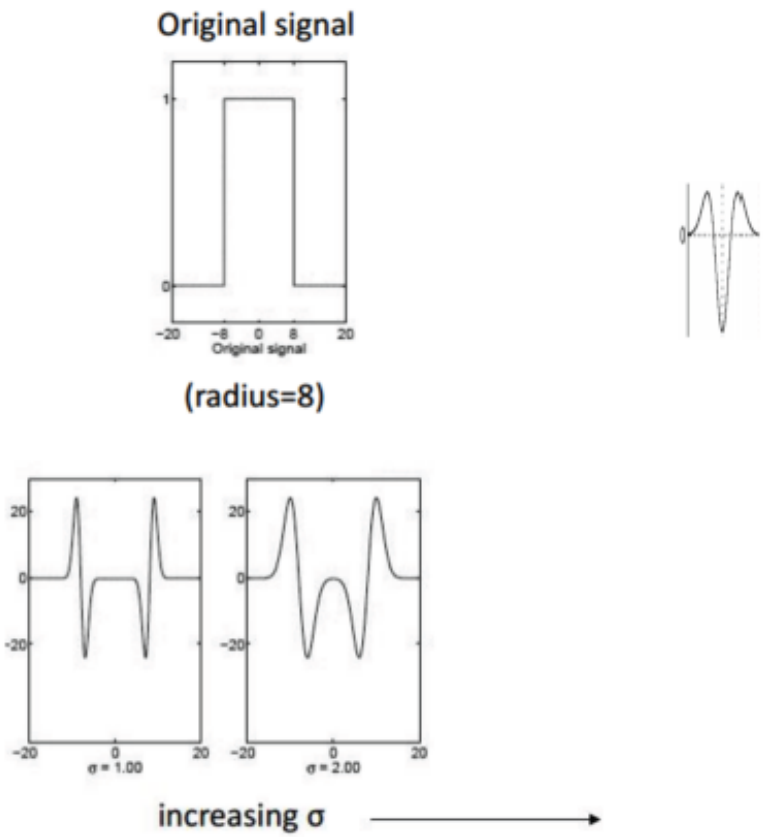
How is it a “blob” detector



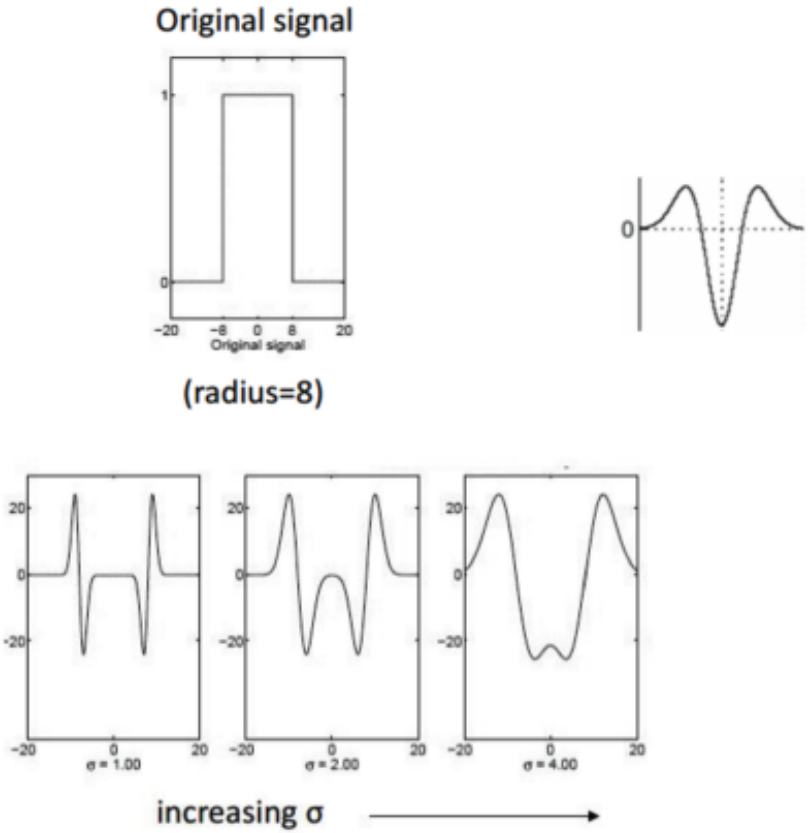
(radius=8)



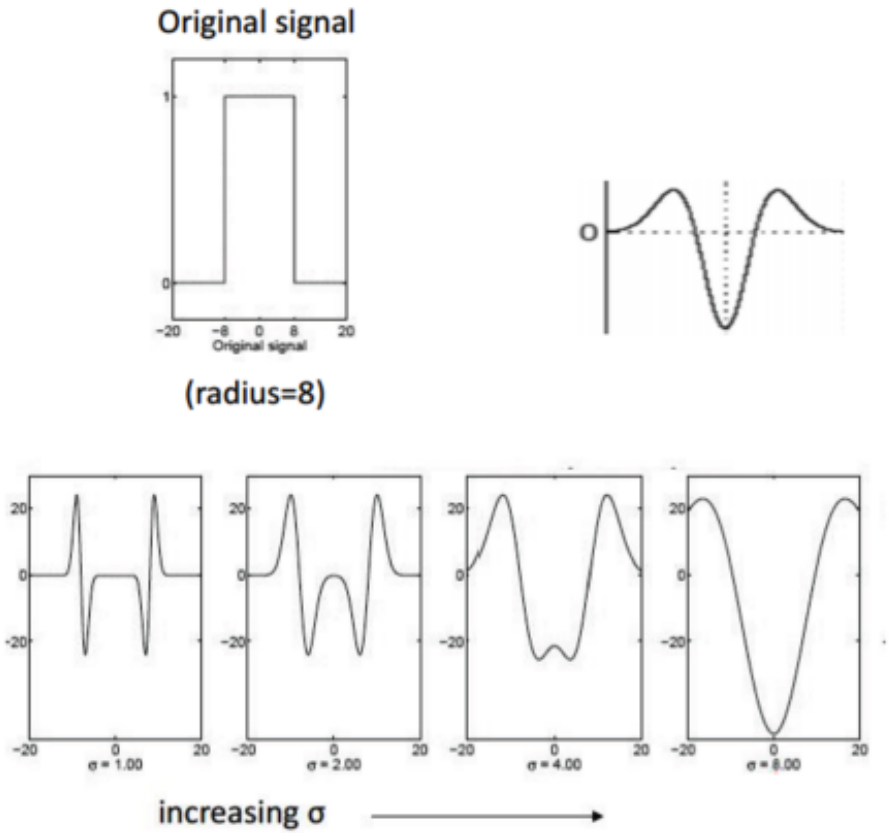
How is it a “blob” detector



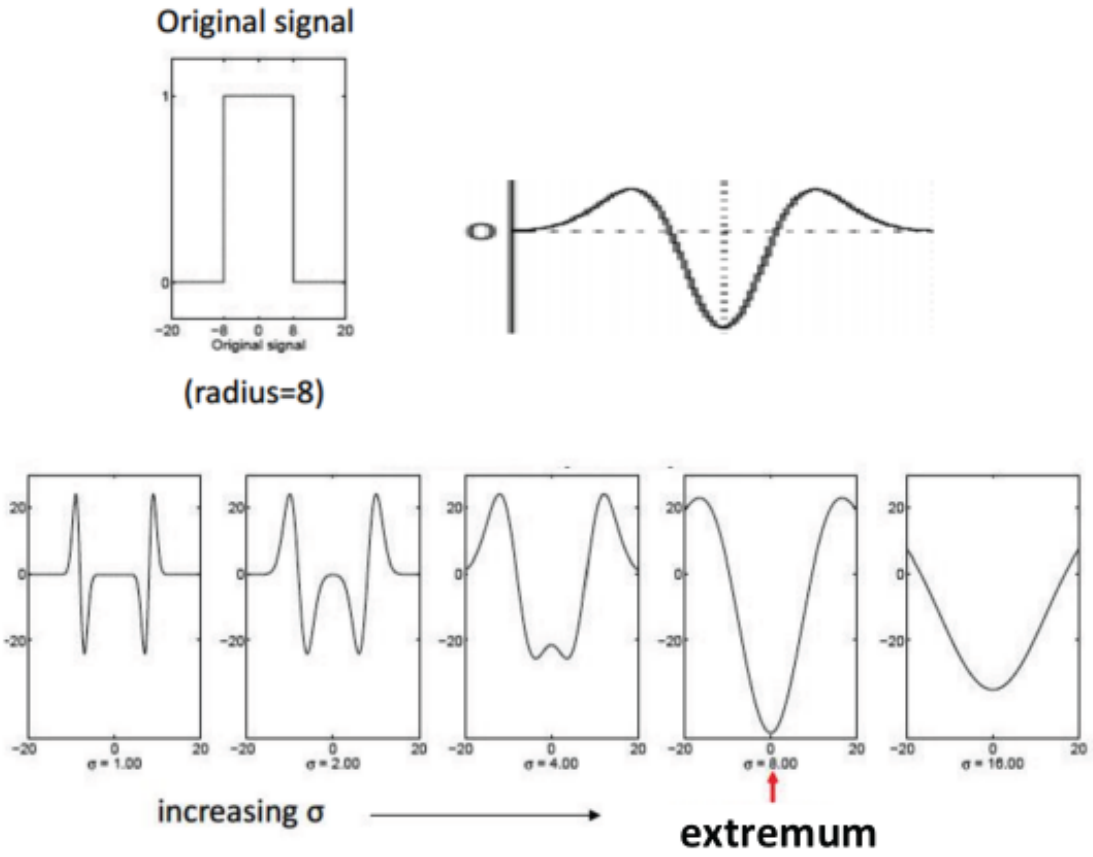
How is it a “blob” detector



How is it a “blob” detector

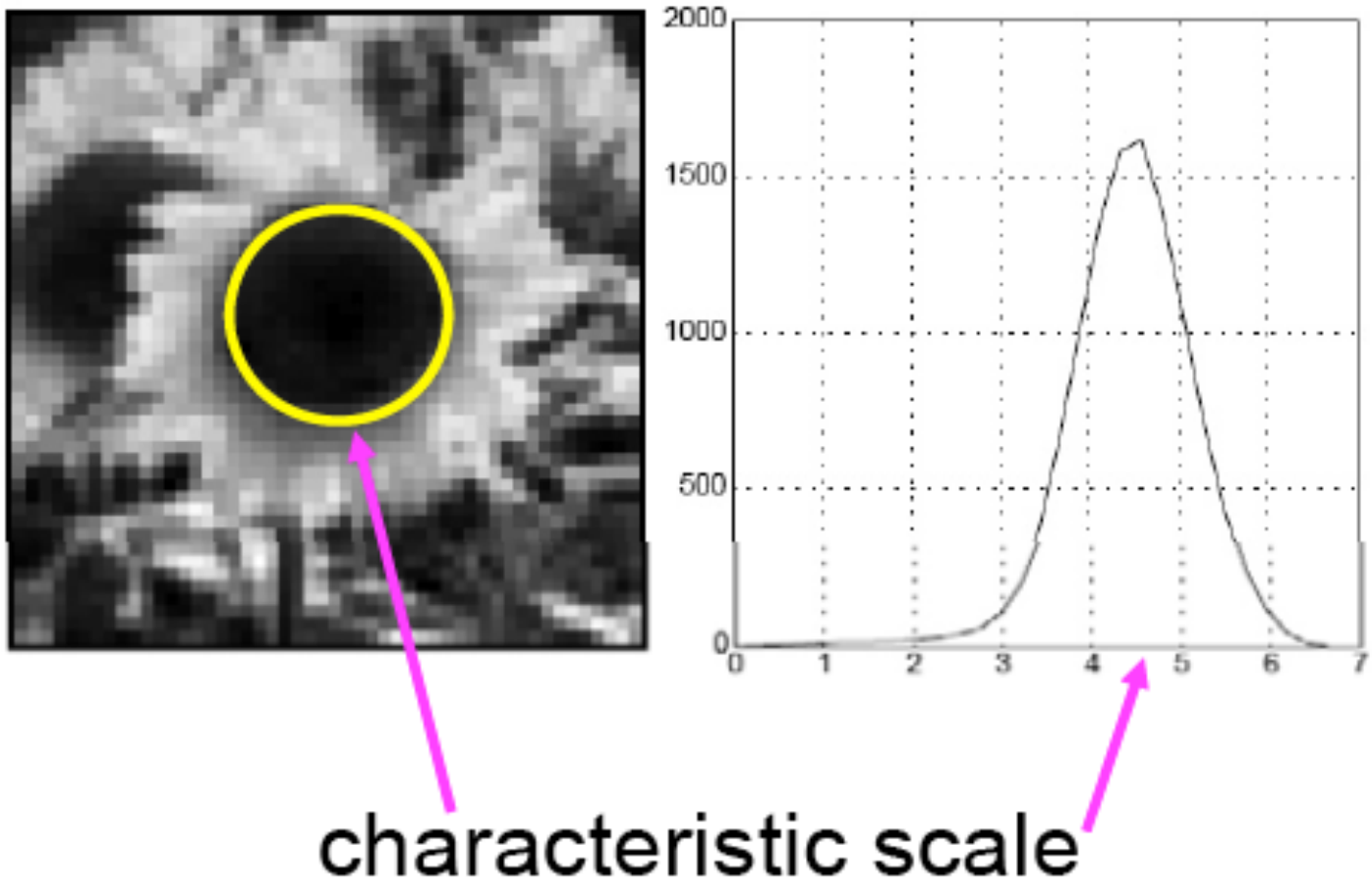


How is it a “blob” detector

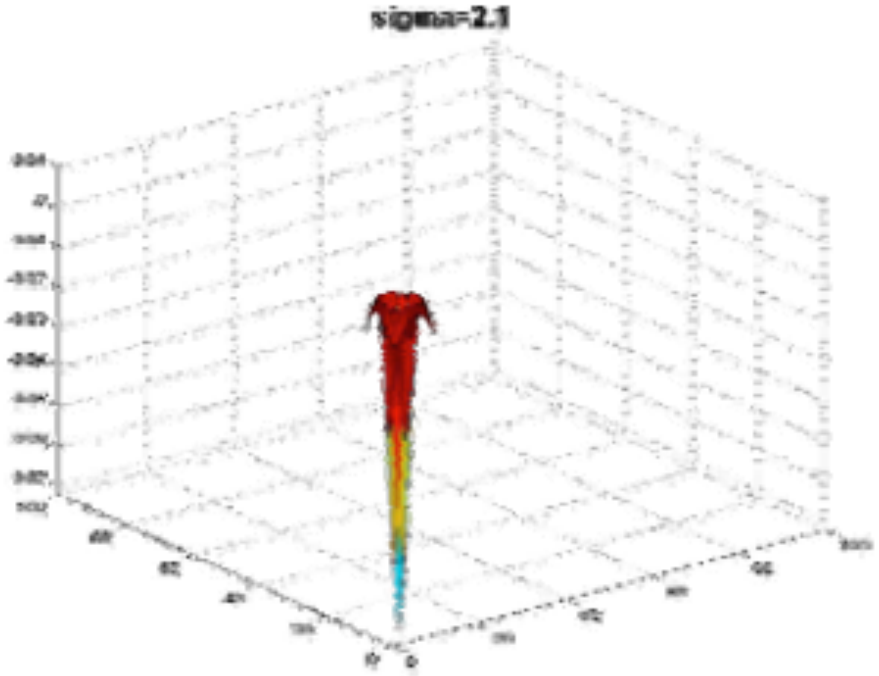


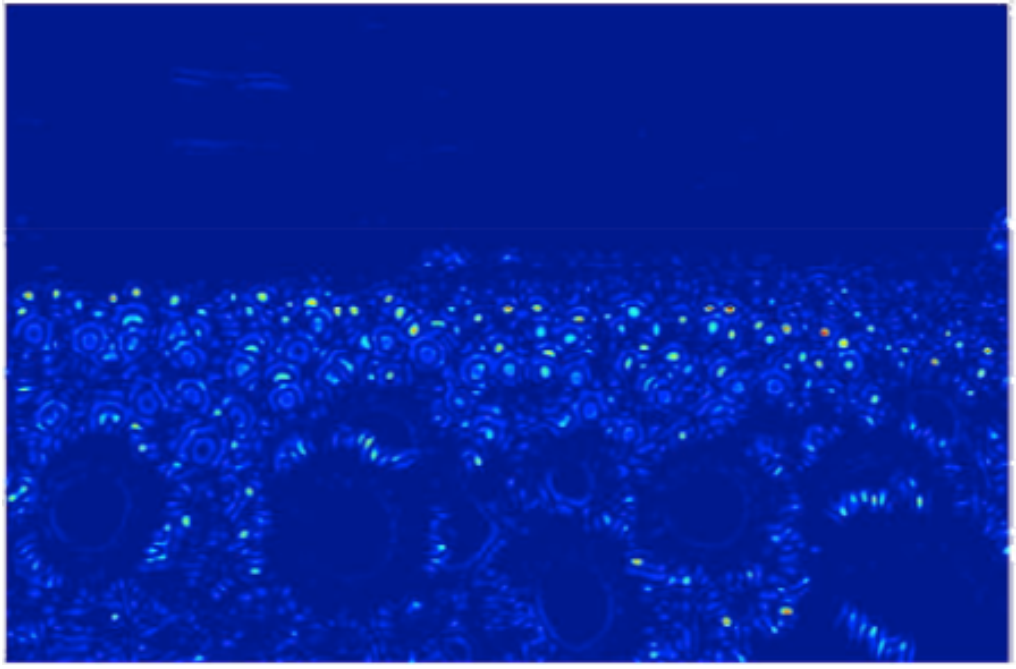
“Characteristic Scale”

Scale that produces a peak of Laplacian Response.

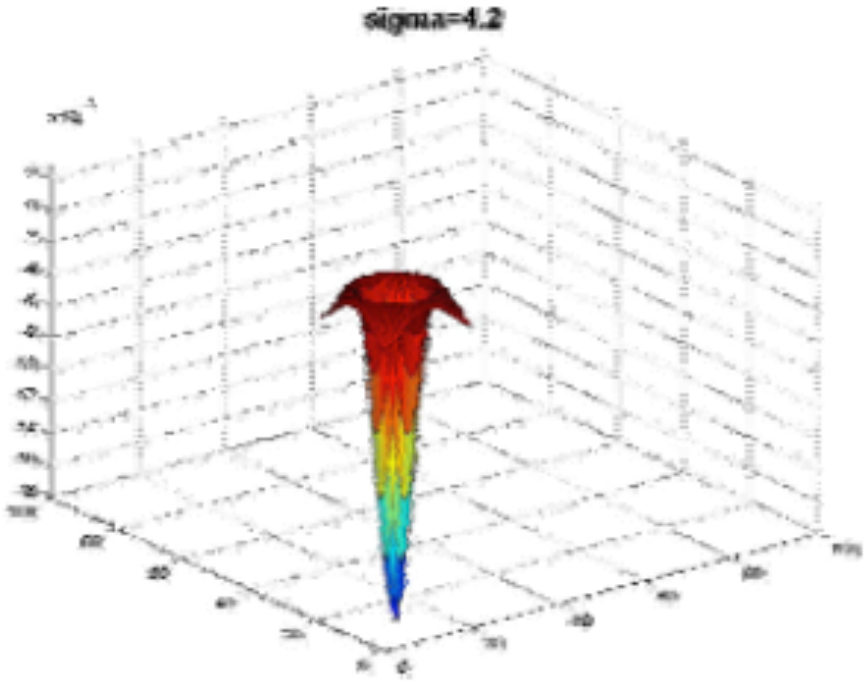


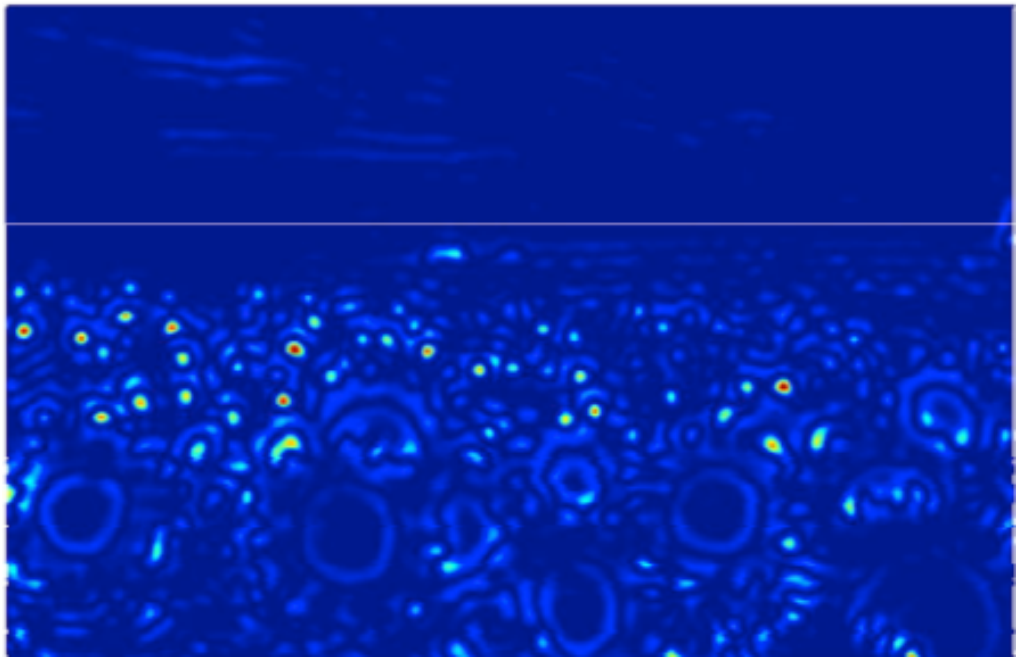
LoG



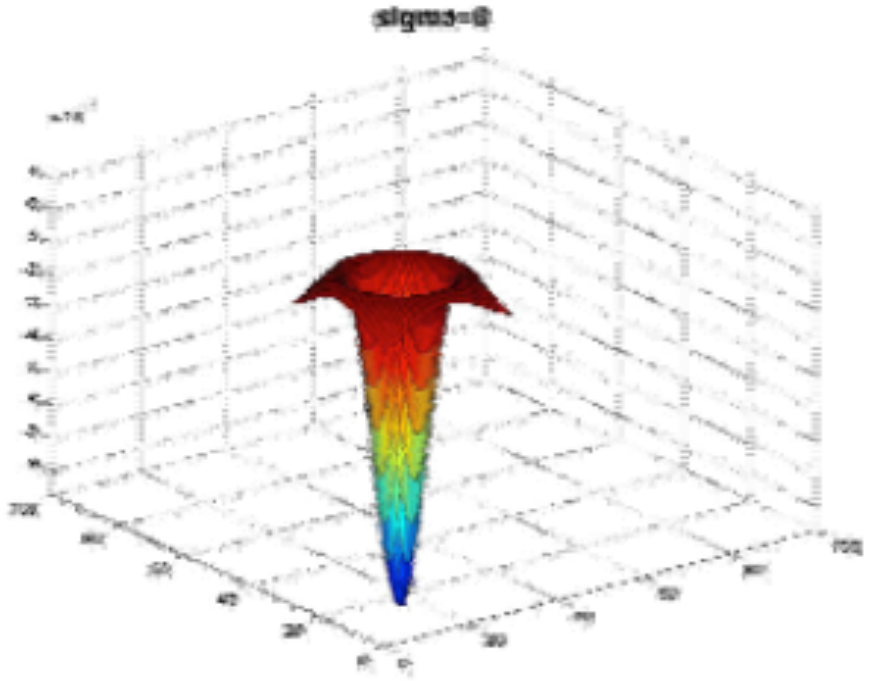


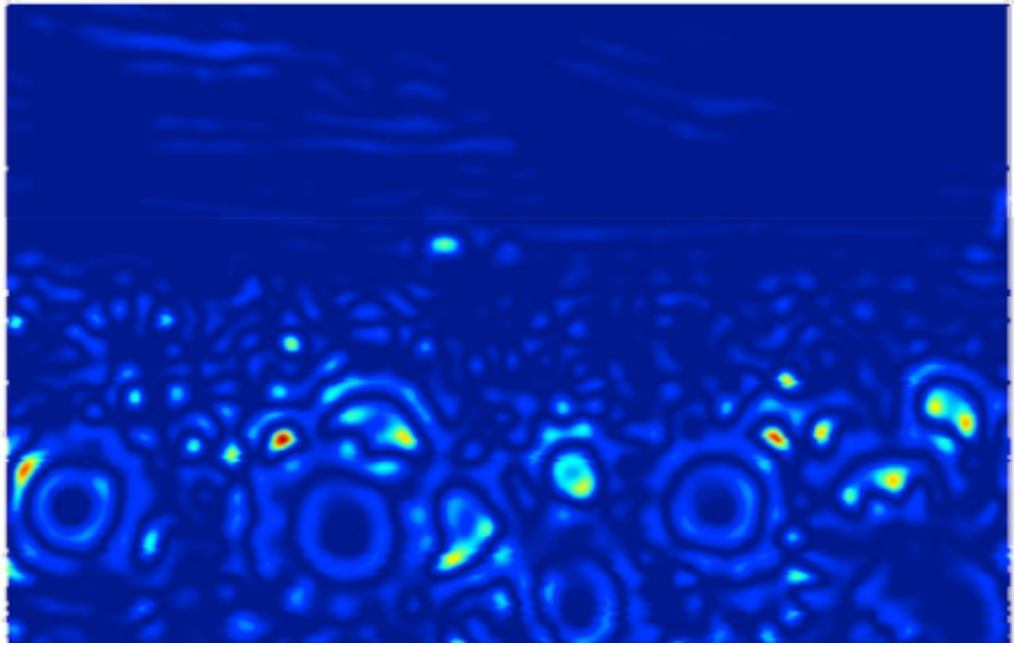
LoG



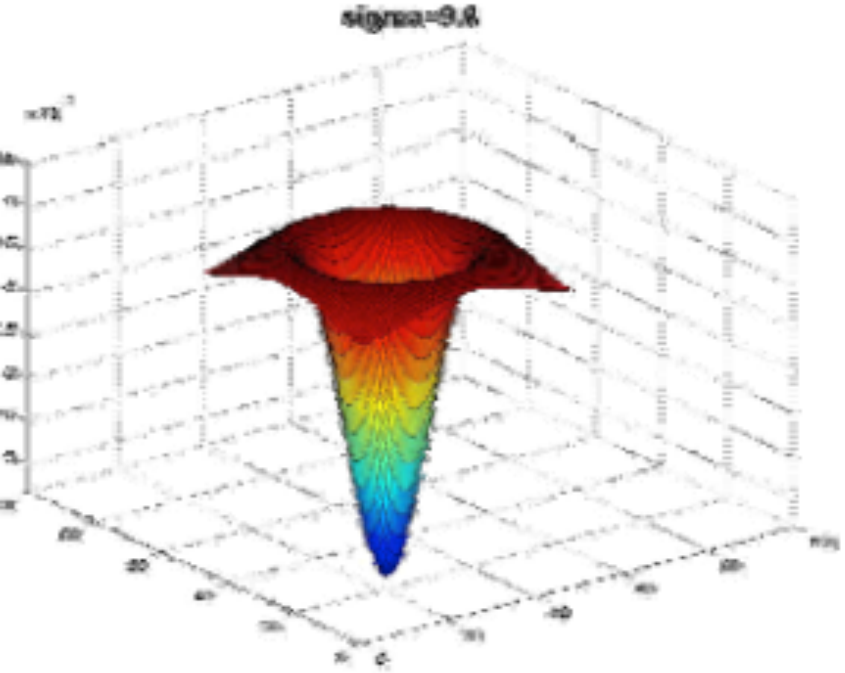


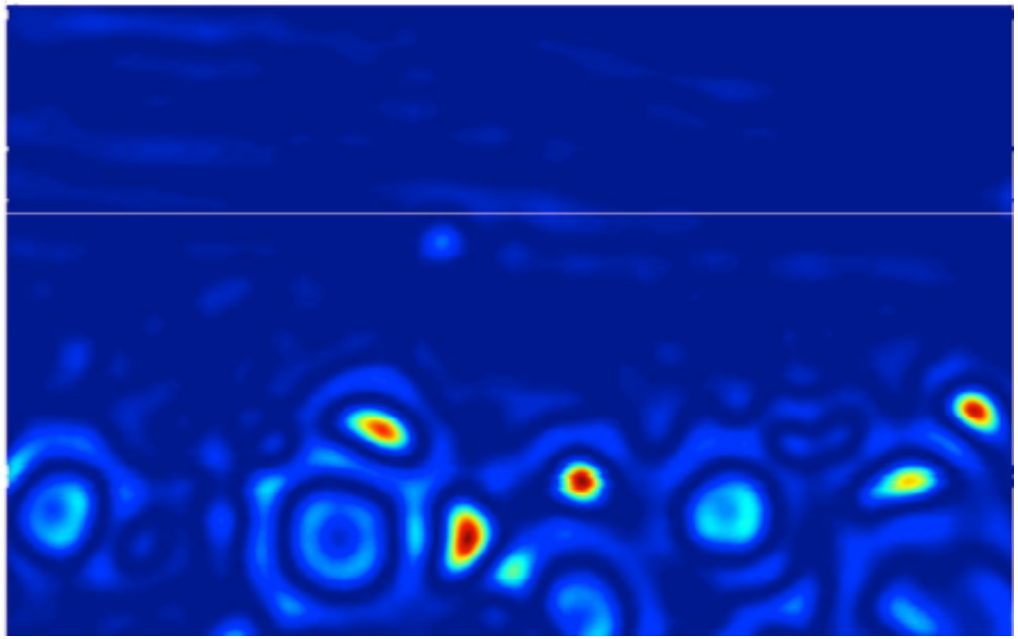
LoG



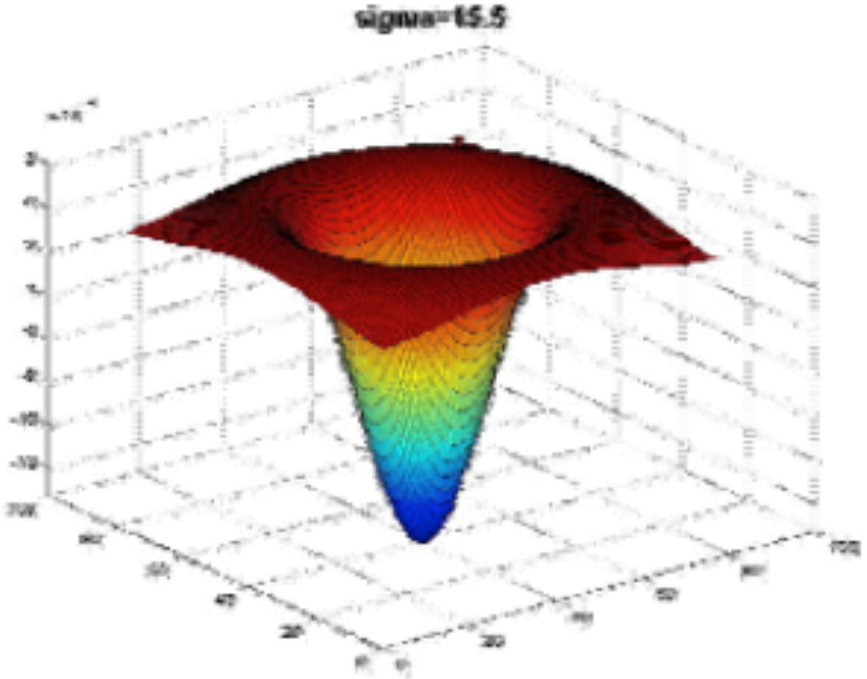


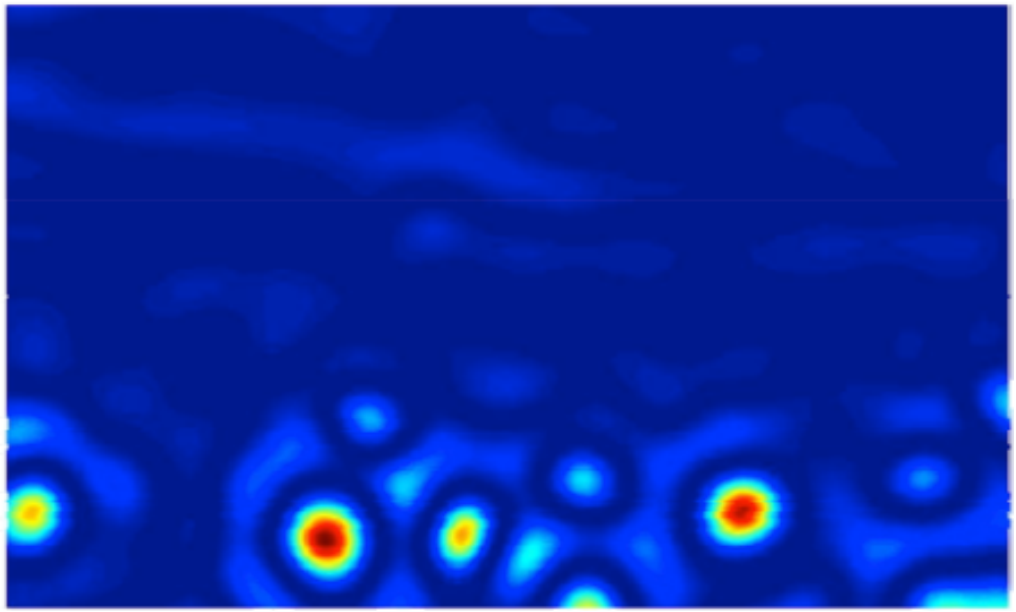
LoG



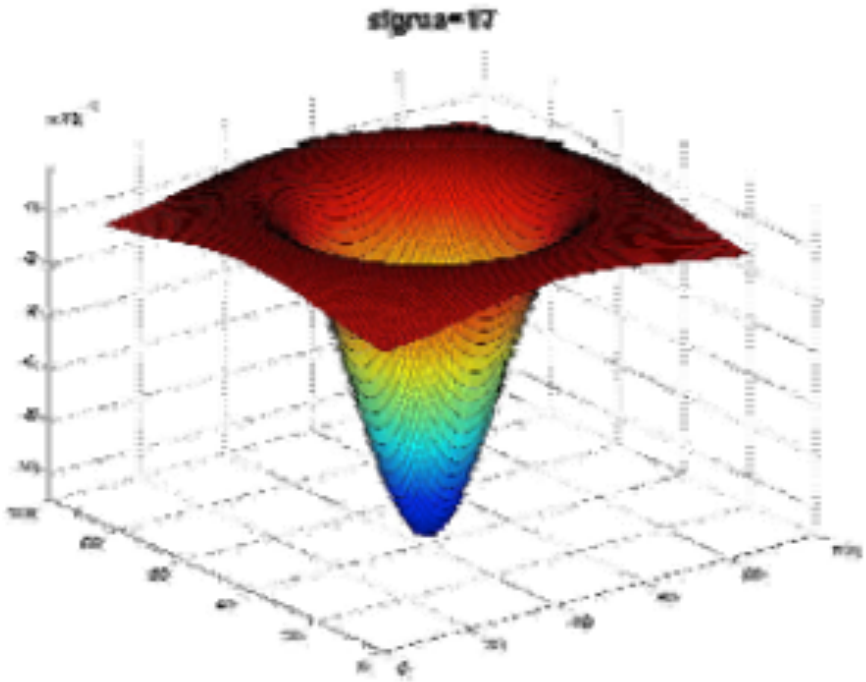


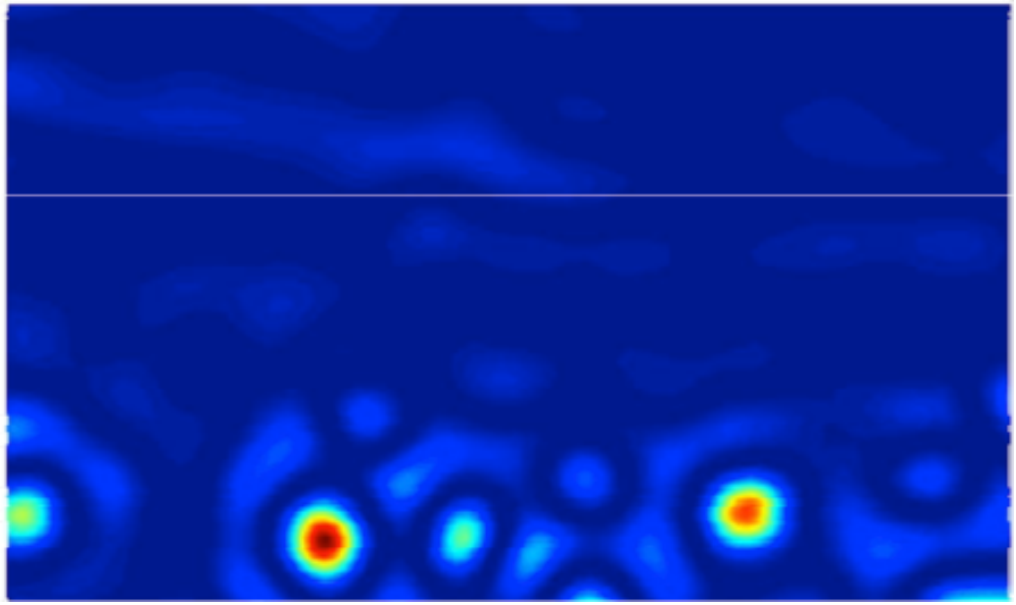
LoG





LoG



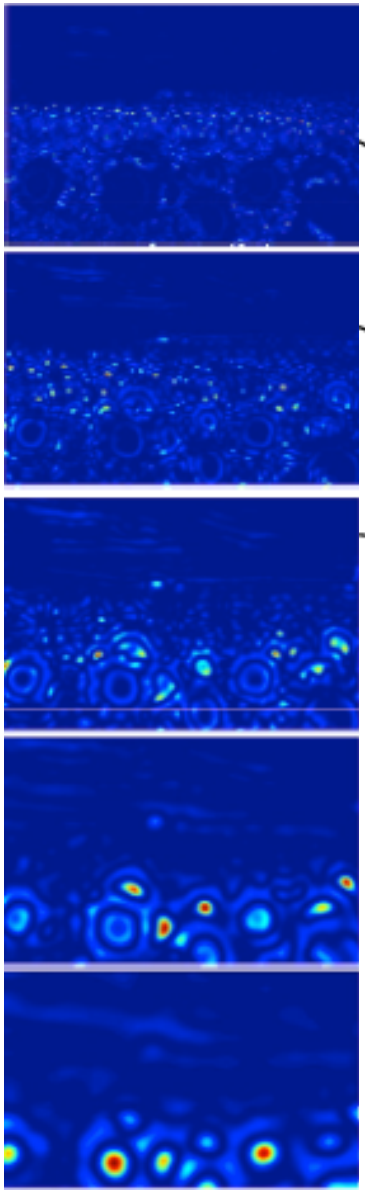


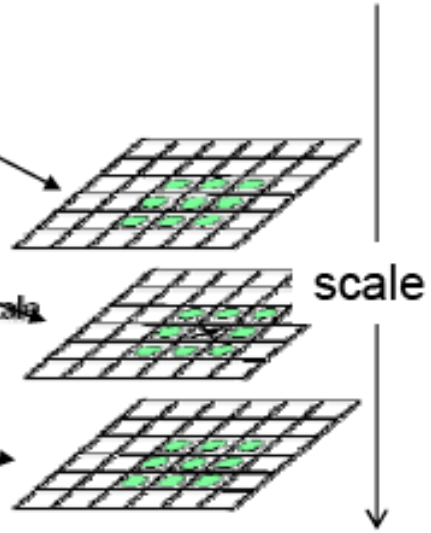
Scale-Space Blob Detection

Interest points are local maxima in both position and scale.

$$L_{xx}(\sigma) + L_{yy}(\sigma)$$

σ5
σ4
σ3
σ2
σ1



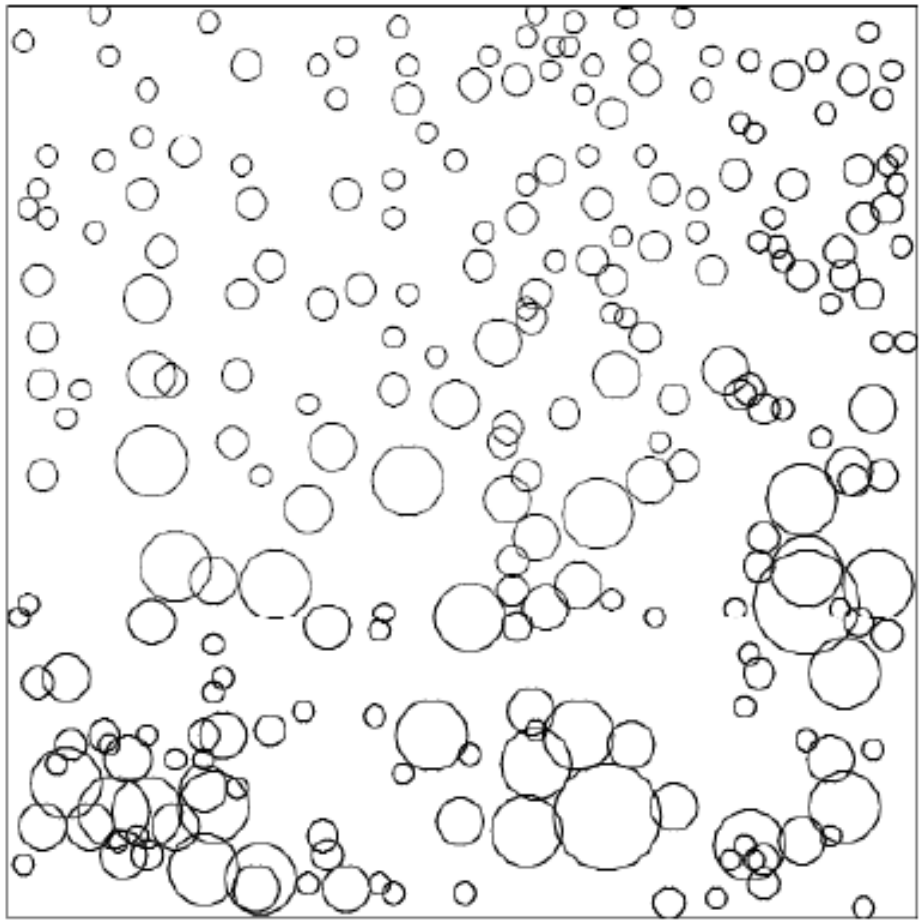


List of x,y, sigma

Example

original image

scale-space maxima of $(\nabla^2_{norm} L)^2$



T. Lindeberg. Feature detection with automatic scale selection. IJCV 1998.

Computation Time

Computing LoG requires convolutions at different scales

Large σ requires large filters (more time to compute)

Scale Invariant Detection

Functions for determining scale

$$f = \text{Kernel} * \text{Image}$$

Kernels:

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

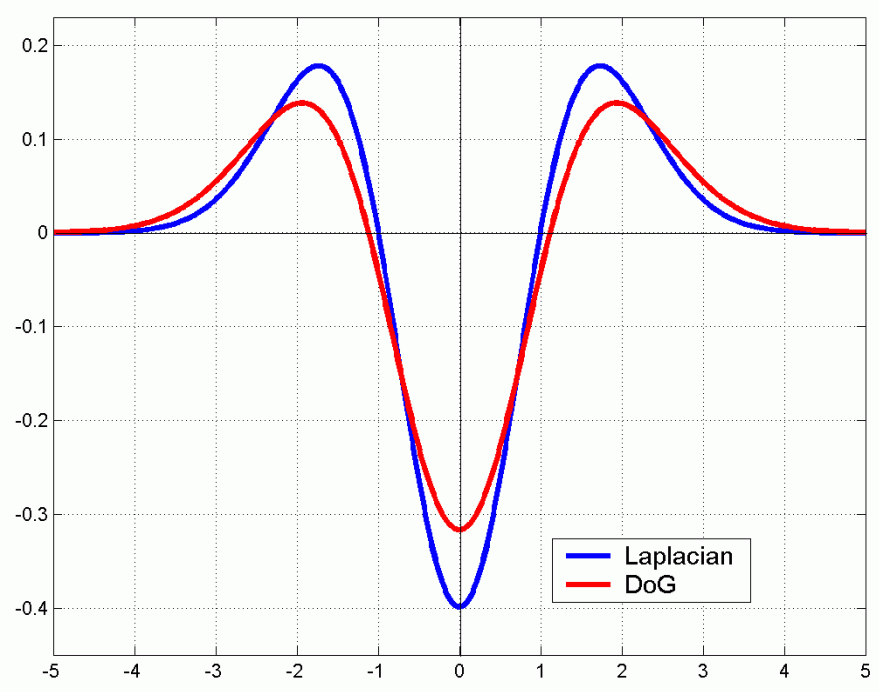
(Laplacian)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)
(more efficient to implement)

where Gaussian

$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$



Note: both kernels are invariant to *scale* and *rotation*

DoG Approximation of the LoG

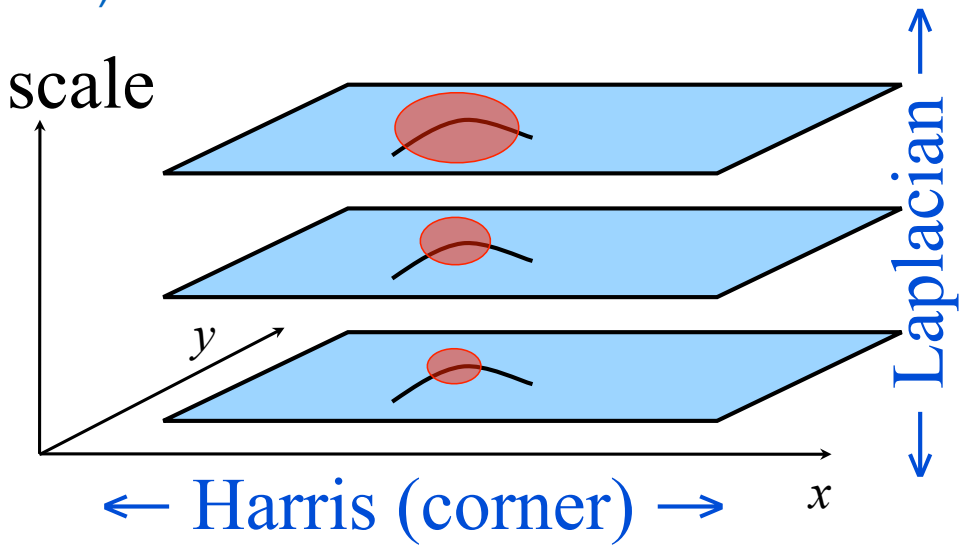
Scale Invariant Detectors

Harris-Laplacian

Find local maximum of:

Harris corner detector in space (image coordinates)

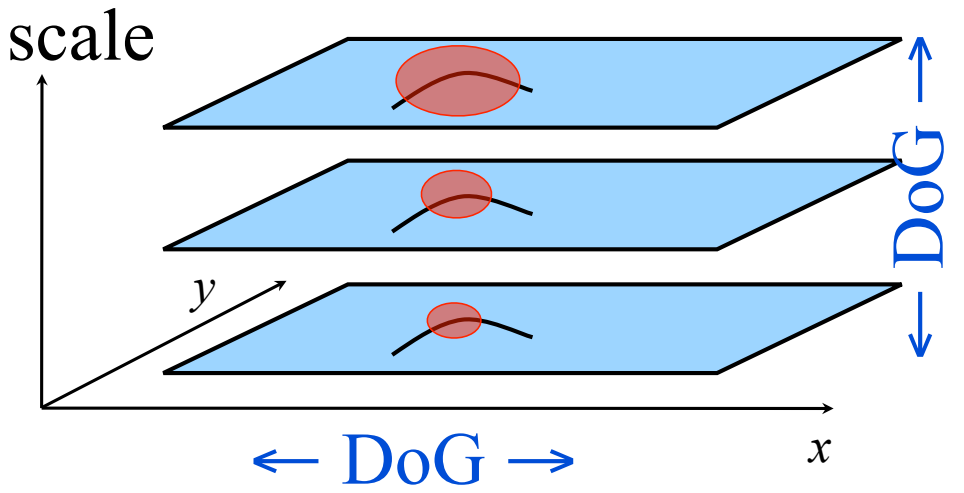
Laplacian in scale



- SIFT (Lowe)

Find local maximum of:

– Difference of Gaussians in space and scale

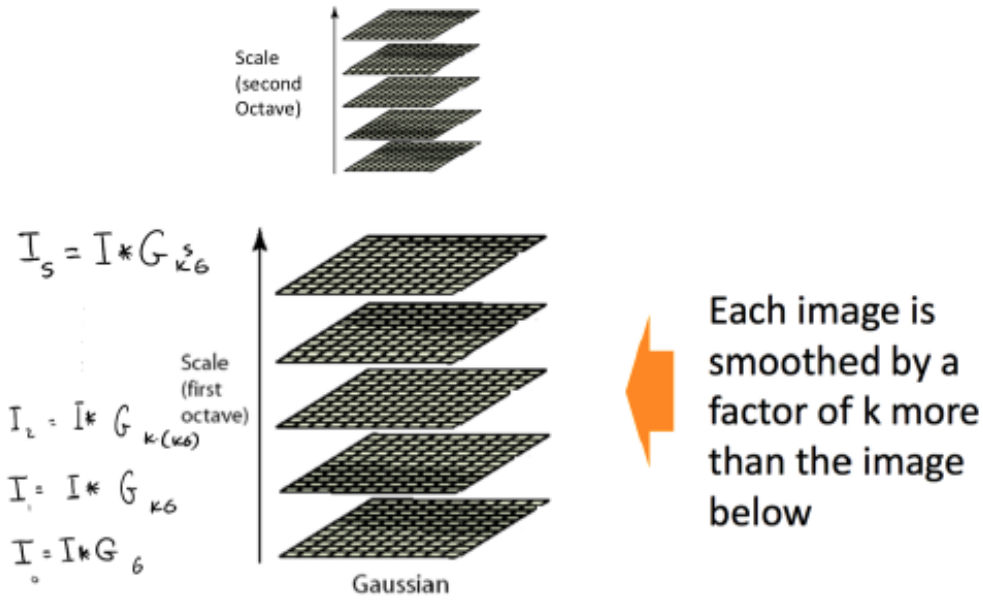


D. Lowe's DoG

Compute a Gaussian Image Pyramid

Compute Difference of Gaussians at every scale

Find local maxima in scale



D. Lowe's DoG

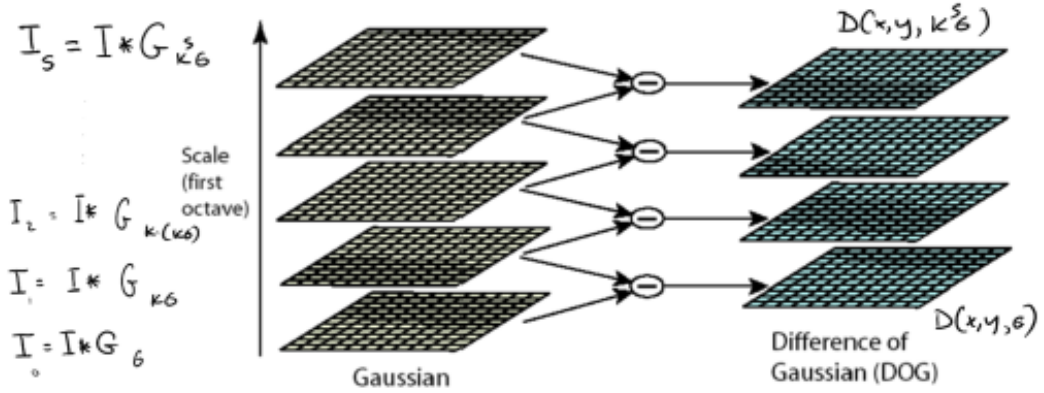
Compute a Gaussian Image Pyramid

Compute Difference of Gaussians at every scale

Find local maxima in scale

$$D(x, y, \rho) = I(x, y) * (G(x, y, k\rho) - G(x, y, \rho))$$

$$\text{for } \rho = \{\sigma, k\sigma, k^2\sigma, \ldots, k^{s-1}\sigma\}, \quad k = 2^{1/s}$$

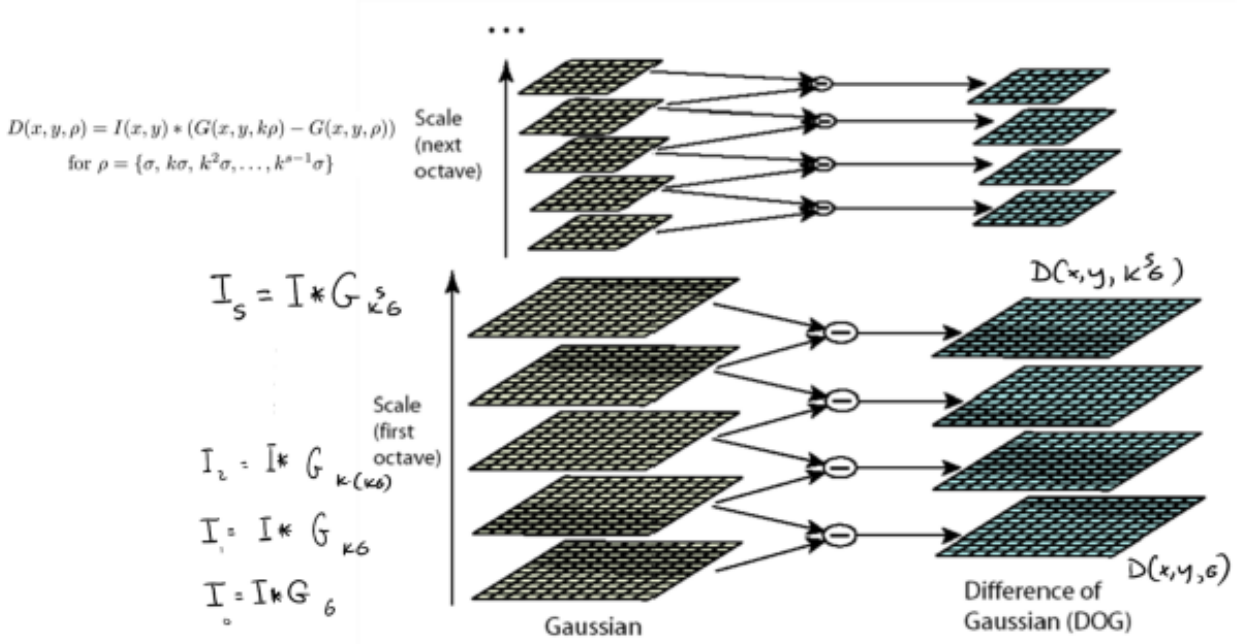


D. Lowe's DoG

Compute a Gaussian Image Pyramid

Compute Difference of Gaussians at every scale

Find local maxima in scale



D. Lowe's DoG

Compute a Gaussian Image Pyramid

Compute Difference of Gaussians at every scale

Find local maxima in scale